2.22 RADAR MODE

The purpose of the Radar Mode Decision Functional Element (FE) is to simulate pilot operation of the radar. The pilot makes decisions relating to the actual operation of the radar set, including radar mode switching, pattern selection and pattern positioning. Simulation of these decisions occurs two ways, through the default pilot decision logic and/or through production rules.

The pilot decision model provides three radar mode selection options, scan, track while scan (TWS) and single target track (STT). Electronically scanned array (ESA) radars are not modeled but can be addressed via production rules. Each alternative has a value function associated with it. These value functions are composed of multiple components, influenced by different situational factors. Each value function is evaluated for the current situation to provide scores for the candidate radar mode selections. The radar mode that scores highest is the one that will be selected by the pilot.

The pilot model also makes decisions regarding selection of the radar scan pattern and pattern position. The default pilot model will select from among predefined TWS patterns when the desired mode is TWS. It will not adjust pattern size parameters (pattern azimuth halfwidth and number of scan bars) for scan and STT modes. Default pattern parameters are used in these cases. The pilot has greater decision making capability for radar pattern positioning. The default behavior is to position the scan pattern so as to cover the target of interest and as many other known hostiles as possible. The pilot will also shift the position of the pattern around the desired direction if the radar fails to make detections within a few frames. If the pilot is attempting to lock up a target using a STT antenna, he will place the pattern center as close to the perceived target line of sight (LOS) to the target as possible. Once lock is achieved, no further pilot input is required. TWS radars automatically place the radar pattern to cover as many high priority targets as possible. If no tracks exist, the pilot must decide where to center the TWS pattern, using logic similar to that used for scan.

The second method for controlling radar mode and pattern is through production rules. Production rules are user written routines that are used to tailor the behavior of the simulated pilots to correspond to study specific requirements. These routines may call existing radar utility routines to set variables such as desired radar mode, desired pattern and pattern position. Values of variables calculated in the production rules may override the values calculated by the simulation, with the caveat that the production rule directive may not be blindly accepted. For example, production rules values may be disregarded if they are not consistent with events currently happening in the simulation (e.g. if production rules direct the pilot to attempt STT lock on a target that is not known to him).

This functional element only models the decisions made by the pilot. Actual changes to the functioning of the radar model itself occur indirectly as follows: Changes to desired radar settings cause the planting of a radar switch change event to simulate the delays associated with physically changing radar controls. Execution of the switch change event actually changes the control inputs of the radar, which then begins responding to those inputs.

2.22.1 Functional Element Design Requirements

The Radar Mode Selection FE will simulate pilot operation of a gimballed (non-ESA) radar. This will be accomplished by simulating decisions for selecting radar mode, pattern and pattern position. This FE will only cover pilot decision making with regards to desired radar control settings. Effects of changes to the radar switch settings and the response of the radar hardware to those changes are a part of the radar functional area and are outside the scope of this FE. Pilot interaction with the newer ESA radar model will also not be discussed in this section because any such interaction must be via production rules; there is no default pilot behavior for ESA radars.

a. The first radar operation decision will be to determine which radar mode to use. The model will simulate two basic radar types, STT and TWS. Decisions related to radar operation will be simulated using one of two methods, the pilot decision model or production rule control. In the case of radars with more than one antenna, the pilot model will only make decisions for the first, or primary, antenna. Management of subsidiary antennas will be done via production rules.

b. An STT radar may operate in either scan or STT mode, the latter of which is entered by obtaining radar lock on a target. The decision to obtain radar lock for STT antennas is made at the pilot posture level for primary antennas or through production rules for both primary and subsidiary antennas. To simulate the setting of switches and the manual movement of the radar, actual lockup is delayed until a minimum time delay has expired. The actual lock up occurs after this time delay if the signal to noise ratio (SNR) exceeds a minimum value on a subsequent successful target detection. Once locked, the radar beam is slaved to the target position, not requiring any further pilot input. No other targets may be detected by the radar while locked.

c. The loss of radar lock due to a number of sources is also simulated. Upon loss of lock, the radar returns to the mode that it was in before it entered STT and will again be managed by the pilot. The pilot may also decide to manually switch from STT mode to scan mode. A pilot decision to break lock will occur if he has locked onto a friendly or if he has chosen another target during his weapon selection decision. Production rules can be used to turn an STT antenna off, put it into scan mode, or attempt to lock up on a specified target in STT mode. Production rules can also be used to put an antenna into close-in-combat (CIC) or velocity search submodes if they are available.

d. A TWS radar may operate in scan, STT or TWS modes. The pilot may decide to switch between scan and STT modes as with an STT radar, or he may choose TWS as his radar mode. The factors that influence the pilot to choose TWS mode are similar to those for choosing scan mode. The differentiating factors that may cause TWS mode to be chosen over scan mode are the existence of target tracks and missiles requiring radar support.

e. The second radar operation decision to be made will be which radar pattern to use. The pilot decision model will be limited to choosing radar patterns when in TWS mode. When changing the TWS pattern the pilot will select the pattern that he believes will cover the greatest number of known hostiles. A default pattern is implemented from the data input file when using scan. The choice of scan radar patterns is greater if production rules are used. Radar pattern parameters can be set for either scan or TWS mode by the production rules.

f. The third radar operation decision is where to place the radar pattern center. Options will exist for using the pilot decision model in scan, TWS and STT modes or using the production rules to set the radar pattern center.

g. In scan mode, the pilot positions the radar pattern depending on availability of GCI information, desire to lock on a target, or in response to failure to make detections in the current position. If the pilot is following a GCI vector, he will center the pattern in the direction indicated by the most recently received GCI message, which will have included a bearing to the target. If the pilot has a specific radar target in mind, the pattern will be centered as close to the pilot's estimate of the target's position as the radar antenna's FOR limits allow. If the pilot cannot quickly establish lock or is having trouble making detections in scan mode, he may shift the radar pattern center about the current setting.

h. In STT mode, the placement of the radar pattern center is an automatic function of the radar hardware, and will not require any pilot input once the radar has locked up. In TWS mode, the pilot will choose the radar pattern placement if no established tracks exist or if he has selected a target but has no TWS track on it. If the pilot has a selected target, he will attempt to place the TWS pattern center on the target's position. If no target has been selected, the pilot places the pattern center the same as he would for scan mode. If at least one track exists in the track bank, then an automated procedure is modeled that attempts to maximize the number of known targets that can be tracked.

i. Production rules may also be used to define the placement of the radar pattern center. Algorithms for pattern placement will be supplied by the user in this case.

2.22.2 Functional Element Design Approach

Pilot operation of the radar can be decomposed into the following design structure.

1. Set the desired radar mode.
    - 1.1 Allow user defined code to set the radar mode.
    - 1.2 Brawler pilot model decides the radar mode.
        - 1.2.1 Pilot evaluates using single target track (STT) mode.
        - 1.2.2 Pilot evaluates using scan mode.
        - 1.2.3 Pilot evaluates using track while scan (TWS) mode, if available.
2. Set the radar pattern.
    - 2.1 Set radar pattern if changing to scan mode.
    - 2.2 Set radar pattern if changing to TWS mode.
    - 2.3 Set radar pattern if changing to STT mode.
3. Set the radar pattern position.

Subroutine *selrdr* acts as the executive routine for this functional element.

Design Element 22-1: Determine Method for Decision

A check is first made to see if the production rules will determine the desired radar mode selection. If the production rules do not control the desired radar mode selection, the pilot will make the radar mode selection decision, in which case value functions are evaluated for each of the three radar modes, STT, scan and TWS. The values are a combination of several components, each an award or penalty for a given situation or condition such as the range to the nearest hostile or the selection of a weapon requiring radar support.

Design Element 22-2: Allow User Defined Code to set the Radar Mode

Subroutine *rrrule* handles the production rules control of radar antennas. Control of the radar antenna may include selection of the desired radar mode, desired radar pattern selection, or selection of the desired radar pattern position. The radar mode desired by the production rules is set unless doing so would result in loss of illumination for a semi-active missile in the air. The production rule request is also ignored if STT has been requested for a target not in the mental model or not perceived to be hostile, unless the CIC submode is active. The STT/CIC combination is allowed, as this simulates an auto-acquisition mode where the radar searches in CIC mode and automatically transitions to STT at the first detection.

Design Element 22-3: Brawler Pilot Decides the Radar Mode

The simulation of the pilot making a radar mode selection decision is contained in subroutine *selrdr* though calls to *rrsttv* and *vscan*, that evaluate the value functions for being in scan, STT and TWS radar modes. Subroutines *rrsttv* and *vscan* are detailed below. After all of the possible modes are scored, the highest scoring alternative becomes the desired mode.

Once the radar mode values have been assigned, the radar mode with the highest score will be chosen by the pilot. The desired radar mode selection is then recorded in the pilot's mental model. Note that this is a recording of desired mode change, not actual mode change. The actual mode change occurs elsewhere in Brawler, after an appropriate delay has elapsed, simulating the time for the pilot to manipulate the radar cockpit instrumentation switches.

The desired radar pattern and pattern position will also be set either through production rules or through the pilot decision model. If they are set through production rules, the values from the rules will be recorded as the desired values in the pilot's mental model. Depending on the selected radar mode, an automatic selection of desired radar pattern position may occur. Automatic selection occurs in STT mode when a target is locked on and in TWS mode if a valid track exists. The pilot decision model will choose the desired radar pattern and pattern position the remainder of the time.

Design Element 22-4: Pilot Evaluates Using Single Target Track (STT) Mode

Subroutine *rrsttv* evaluates the value function for locking up in STT on a specified aircraft. The value function is evaluated for each hostile known to the pilot. The value function for attempting to lock up in STT is influenced by the following factors:

j. Hysteresis - is the target the current lock choice? The amount of points accrued depends upon whether or not lock has been achieved.

k. Estimated ability to achieve lock. This is a function of the estimated chance of acquiring the target due to uncertainty in target elevation angle and elevation angular rate.

l. A reward for the candidate being the current target of a weapon requiring radar guidance.

m. A bias against targets that may exceed the field of regard limits.

n. Consistency with the weapon-target selection.

o. Consistency with flight tactic orders.

Design Element 22-5: Pilot evaluates using scan mode

Subroutine *vscan* evaluates the value function for being in radar scan mode. The scan value function consists of two components. These components are influenced by range to hostiles and scan volume.

The value of the scan alternative is primarily a function of range to hostiles. At long ranges the scan value is always very high. At shorter ranges the value functions have been tailored so that the scan value exceeds the value of a lock alternative when the candidate target is still at least ten seconds from presenting a firing opportunity. The motivation is to stay in scan mode to gather information on other aircraft for as long as possible.

The scan mode alternative receives an additional scoring component that attempts to capture the ‘information’ value for being in scan mode versus STT and TWS modes. This additional value is based on the volume of the applicable scan pattern. This term will give added weight to the scan alternative if its scan volume is larger than that for TWS, while the antenna is in TWS mode. The longer the pilot remains in TWS mode, the larger this additional score for scan becomes. The motivation being that the pilot is denying himself information by remaining in the smaller scan volume associated with TWS mode. Similarly if the pilot is in STT mode, both TWS and scan will receive additional ‘information’ score since their scan volumes are larger than the STT scan volume.

Design Element 22-6: Pilot Evaluates Using Track While Scan (TWS) Mode

Subroutine *vscan* also evaluates the value function for being in TWS mode. The TWS value function consists of three components. These components are influenced by the following factors:

a. The range to known hostile targets.
b. The size of the scan volume, compared to scan volume size of other modes.
c. The number of tracks already established in the track bank.
d. Whether or not currently selected weapon requires STT support.
e. Whether or not currently selected weapon will be launched command guided.
f. Existence of a track on the currently selected target.
g. Orders to attack a specific aircraft.

h. Missiles in the air that require data link updates.

The first value function component for being in TWS mode is set equal to the first component of the scan value function, based on range to hostiles. The second TWS value function component is a function of scan volume (see scan value function component 2 in the previous design element description). The third component may receive separate awards due to each of the factors c-h listed above. The motivation for the first two components is the same for the first two components of the scan value function. The motivation for awarding value for the third component is to bias TWS over scan mode when target tracks exist and missiles require support.

Design Element 22-7: Set Radar Pattern if Changing to Scan Mode

Subroutine *dessca* records a pilot's intent to place a radar in scan mode. When the desired mode is scan, the desired pattern width and number of bars are set to the default values for this antenna. The pattern azimuth and elevation are also set to their default values, with the expectation that they will be reset in subroutine *rdrelv*, if necessary. These default values are read in from the **SCNRIO** file during initialization and may be different for each antenna and each radar. If production rules control the radar pattern or pattern position, the values from the production rules will be used in place of the default values.

Design Element 22-8: Set Radar Pattern if Changing to TWS Mode

Subroutine *destws* records a pilot's request to place a radar in TWS mode. When the desired mode is TWS, the pilot will select the TWS pattern that he believes will cover the largest number of hostiles. The pattern azimuth and elevation are reset to their default values, with the expectation that they will be reset in subroutine *rdrelv*, if necessary. If production rules control the radar pattern or pattern position, the values from the production rules will be used in place of the default values.

Design Element 22-9: Set Radar Pattern if Changing to STT Mode

Subroutine *desstt* records a pilot's intent to place a radar in STT mode. Entering STT mode is modeled as requiring a finite time to complete, due to switchology and acquisition delays. The average time required to achieve lock, given an adequate signal return from the target, is part of the antenna characteristics data and is read in during initialization. This functional element just records the pilot’s intent to go into STT and computes the earliest time at which lock can be achieved.

Design Element 22-10: Set Radar Position

The algorithm used to determine the desired azimuth and elevation of the center of the radar scan or TWS pattern depends upon the desired mode for the radar. If an antenna is to be put into scan mode, the positioning of the radar pattern will be determined by several factors. If the pilot is being vectored by GCI/AWACS controllers, the pattern will be centered along the target bearing received from the controller. If the pilot desires to lock up in STT or has selected a target, the pattern will be centered on the pilot’s estimate of the target’s position. In either case, a test is performed to determine if the desired direction is within the antenna’s gimbal limits. If so, that direction is used. If not, the pattern is centered against the gimbal limit as close as possible to the desired direction. Lacking these controlling factors, the pattern will point at the azimuth and elevation default values specified in the **SCNRIO** input file. Brawler also simulates the fact that pilots will not search their selected portions of the sky indefinitely, but will occasionally move the scan pattern. The pattern will also be moved if the pilot is attempting to center the pattern on a chosen target and fails to make a radar detection within two frame times.

If an antenna is already in STT mode and the target being tracked is within the gimbal limits, the radar is automatically positioned to point at the target and no decision by the pilot is required.

If an antenna is in TWS mode, positioning of the pattern is done by the pilot when the radar first enters TWS, since there are no TWS tracks at this time. Once tracks have been established, the radar will automatically center the pattern. If the pilot has selected a target, that target is outside of the currently established TWS pattern, and no SFD or TWS track of the target exists, the pilot will switch the radar into manual TWS positioning. The pattern will be positioned centered on the target's perceived position. Once a track has been achieved, the radar will again automatically position the scan pattern. TWS pattern positioning can also be controlled through production rules.

This functional element is implemented under subroutine *rdrelv*.

2.22.3 Functional Element Software Design

This section contains the software design that implements the functional element requirements and the design approach defined in the preceding sections. The first subsection describes the subroutine hierarchy and describes how the subroutines work to make the radar operation decisions. The remaining subsections contain functional flow diagrams and describe all of the significant operations represented by each block in the diagrams.

Radar Selection Subroutine Hierarchy and Description

The major routines comprising the radar selection algorithm and their purpose are given below with the indentation of the routine name used to indicate the level of the routine within the calling tree.

selrdr- radar mode decision executive

- *rrrule* - handles production rule directives for radar control
- *rrsttv* - evaluates value function for being in STT mode
- *vscan* - evaluates value functions for being in scan and TWS modes
- *dessca* - records a pilot’s request to place a radar in scan mode and selects the radar pattern
- *destws* - records a pilot’s request to place a radar in TWS mode and selects the radar pattern
- *desstt* - records a pilot’s desire to lock up his radar in STT mode
- *rdrelv* - positions aircraft radar antenna in azimuth and elevation

A number of secondary and utility subroutines are used in the situation update process as described below.

anytrk	Determines if the specified avionics device has a track on the specified target.
asstgt	Gets target assigned to pilot by flight leader.
cicang	Creates a new radar frame rotated 90 degrees about the x-axis for use in CIC mode.
ckrngi	Checks to see if a variable is in the allowed range.
cross	Calculates the cross product of two 3-vectors.
des_rdr_off	Records a pilot's desire to turn a radar antenna off.
ftkany	Fills the /*tracks*/ data structure with the contents of the trackbank of the specified device, or SFD if the aircraft is carrying one.
gcum	Evaluates a normal distribution function.
getrhe	Gets the Earth to heading system rotation matrix.
gmisld	Retrieves missile characteristics data.
grdrc	Gets radar characteristic data for the conscious pilot's aircraft.
grdrs	Gets radar status data for the conscious pilot's aircraft.
indupk	Unpacks 4 arguments from a single alternative descriptor word.
lockan	Determines if a given radar antenna is locked on a specified target.
markov	Model that determines if a pilot will shift the radar pattern position.
match	Finds match of element to a list of numeric elements.
mystor	Returns the number of each weapon type for a given aircraft.
nabort	Aborts run and prints out error messages.
olistv	Places record in list memory to a format not needing to remember the record length.
plantm	Schedules an event at the indicated time.
plt_rdr_swch	Plants a radar switch change event when the pilot wishes to change the radar switch settings.
prnpos	Prints relative position of aircraft to the log file.
rdr_cen_tgt	Computes a radar pattern centered over a target.
rdrlos	Converts azimuth and elevation in radar heading frame into LOS vector in Earth frame.
twspat	Selects the best TWS pattern from the available set.
udrbe	Rotates a matrix about an arbitrary axis.
vecinc	Adds multiple of one vector to another.

vmax	Finds the largest element of an array.
vnorm	Generates a unit vector in the desired direction.
vpoint	Generates a unit vector along a given elevation and azimuth or visa versa.
vsub	Performs a simple vector subtraction.
vxfrmc	Transforms vectors as specified by an orientation matrix.
xmit	Moves or initializes a block of numeric data.

The principal data structures (common blocks) involved in the radar selection process are described below.

/*antsta*/	Holds non-pilot-controllable elements of the radar status for a single radar antenna.
/*extst*/	Stores the external status of all aircraft.
/*fcstat*/	Holds local fire control related variables for a single aircraft.
/*mind3*/	Holds the pilot's perception of physical states and assessment of relationships with other entities.
/*mind3a*/	Stores error information on observed aircraft.
/*mindms*/	Holds mental model perception of missiles.
/*mindpr*/	Holds production rule variables that influence pilot decisions.
/*mindr*/	Stores mental model variables associated with pilot's radar.
/*ppost*/	Holds results of the pilot posture decision.
/*rdrdat*/	Holds reference data for a single aircraft radar device.
/*rdrsta*/	Holds the radar internal status.
/*rdrsws*/	Holds the radar switch status, i.e. pilot controllable parts of the radar status.
/*rrfixd*/	Holds the fixed radar data for a single antenna.
/*rrsttd*/	Holds the STT data for a single antenna.
/*rrtwsd*/	Holds the TWS data for a single antenna.

Subroutine *selrdr* is the executive that handles the simulation of pilot operation of a radar. The pilot decision model will determine pilot choices for radar mode, pattern and pattern position. Subroutine *selrdr* first calls subroutine *rrrule* to determine if the production rules will set the radar parameters in place of the pilot decision model. If the production rules do not control the radar mode, then the value functions are evaluated to determine which radar mode the pilot desires. Subroutine *rrsttv* is called to evaluate the value function for being in STT mode. The value functions for being in scan and TWS modes are evaluated by calling subroutine *vscan*. Once a radar mode has been selected, either by the pilot decision model or through production rules, the desired radar mode is recorded in the pilot's mental model. Subroutine *dessca*, *destws* or *desstt* is called to record the radar mode selection for scan, TWS or STT mode respectively. Subroutines *dessca* and *destws* also determine the pilot's choice for desired radar pattern. Subroutine *rdrelv* is called to decide where the radar pattern center should be positioned. *Selrdr* finishes by calling subroutine *plt_rdr_swch* to plant a radar switch event if the desired radar mode, pattern or pattern position is not the same as the present radar settings.

Subroutine SELRDR

Subroutine *selrdr* is the executive routine for radar mode decisions. *Selrdr* selects a desired radar mode, pattern and pattern position. Figure 2.22-1 is the functional flow diagram that describes the logic used to implement *selrdr*. The blocks are numbered for ease of reference in the following discussion.

<u>Block 1.</u> Variables to be used for indexes to refer to scan or TWS radar modes are initialized based on the number of hostiles on the 'bad guys' list. References to STT mode later use the number of 'bad guys' as an index.

<u>Block 2.</u> Test whether the production rules target flag, *prdtgt*, was set in the production rules. If yes, go to Block 3. If no, go to Block 4.

<u>Block 3.</u> Set the avionics designated target equal to the production rules target. The automatic TWS pattern positioning algorithm moves this aircraft to the top of the list of priority targets. The TWS pattern is positioned to cover as many high priority targets as possible, so moving this target to the top of the list will ensure that it will be inside the TWS pattern.

<u>Block 4.</u> This block represents the top of a DO loop over the number of antennas on this aircraft's radar.

<u>Block 5.</u> Subroutines *grdrc* and *grdrs* are called to get the characteristics and status data for the current antenna in the DO loop.

<u>Block 6.</u> Test whether the antenna is off and not controlled by production rules. If true, go to block 4 to process the next antenna. If false, continue to Block 7.

<u>Block 7.</u> Test whether production rules are requesting the use of LPI. If true, set LPI to the mode requested, otherwise, set LPI to zero.

<u>Block 8.</u> Set the LPI mode equal to the value requested in the production rules. Possible modes are *mindet* and *rwrdny*. In *mindet* mode, the radar tries to use the minimum power required to produce a 50% probability of detection by the target's RWR. In *rwrdny* mode, the radar tries to use the maximum power that may be used without triggering the RWR on the target aircraft.

<u>Block 9.</u> Subroutine *rrrule* is called to execute user defined code (production rules).

<u>Block 10.</u> Test whether production rules controls this antenna. If true, the production rules have set a radar mode, so skip to block 20 to begin processing the mode selected. If false, blocks 11-19 are executed to choose a radar mode. Mode selection is determined in this case by evaluation of value functions for each candidate radar mode. The radar mode that scores the highest based on the value functions becomes the desired radar mode.

Block 11. Test whether this is the primary antenna. If true, continue processing this antenna. Otherwise, proceed to block 4 to process the next antenna (only production rules may currently process non-primary antennas).

Block 12. Test whether there are any hostiles on the 'bad guy' list. If true, execute blocks 13-17 to evaluate the STT value function for each 'bad guy'. If not, jump to Block 18.

Block 13. Subroutine *mystor* is called to retrieve the list of the stores on the conscious pilot's aircraft.

Block 14. Subroutine *asstgt* is called to get the conscious pilot's assigned target from his flight leader.

Block 15. This block represents the top of a DO loop over the number of hostile aircraft on the 'bad guy' list.

Block 16. Subroutine *rrsttv* is called to calculate the value of locking in STT on the hostile. Subroutine *rrsttv* is detailed below.

Block 17. Test whether there are more hostiles on the 'bad guy' list. If true, proceed to the top of the loop (block 15) to process the next hostile.

Block 18. Subroutine *vscan* is called to calculate the value of being in scan mode and the value of being in TWS mode. Subroutine *vscan* is detailed below.

Block 19. Subroutine *vmax* is called to determine which of the candidate radar modes scored highest. It takes a list of the values (one for STT on each hostile, and one each for being in scan or TWS mode) and returns an index to the largest value. This index is used to indicate the desired radar mode chosen (STT, scan or TWS). Once the desired radar mode has been recorded in the pilot's mental model (blocks 21-28), the actual radar mode will be changed elsewhere in Brawler after the appropriate time delay for switch setting has elapsed.

Block 20. At this point a desired radar mode has been chosen, either through production rules or through pilot selection. Test whether the chosen radar mode is OFF. If true, record in the pilot's mental model the pilot's desire to turn off the radar (Block 21). Otherwise, jump to Block 22.

Block 21. Subroutine *des_rdr_off* is called to record in the pilot's mental model the pilot's desire to turn the radar off. The radar can only be turned on or off through production rules, so this decision is only made in response to user input.

Block 22. Test whether the selected radar mode is scan. If true, record the pilot's desire to change the radar mode to scan mode into his mental model (Block 23). Otherwise, jump to Block 24.

Block 23. Subroutine *dessca* is called to record in the pilot's mental model his desire to change the radar mode to scan. Subroutine *dessca* is detailed in one of the following sections.

Block 24. Test whether the selected radar mode is TWS. If true, record in the pilot's mental model his desire to change the radar mode to TWS (Block 25). Otherwise, jump to Block 26.

Block 25. Subroutine *destws* is called to record in the pilot's mental model his desire to change the radar mode to TWS. Subroutine *destws* is detailed in one of the following sections.

Block 26. Test whether the chosen radar mode is STT. If true, record in the pilot's mental model his desire to change the radar mode to STT (Block 28). Otherwise, an invalid mode has been selected, so abort the run and print diagnostic messages (block 27).

Block 27. Subroutine *nabort* is called to abort the run due to and invalid radar mode selection.

Block 28. Subroutine *desstt* is called to record in the pilot's mental model his desire to change the radar mode to STT. Subroutine *desstt* is detailed in one of the following sections.

Block 29. Now that the desired radar mode selection has been recorded in the pilot's mental model, check for changes in desired radar pattern position. Subroutine *rdrelv* is called to set the desired radar pattern position. Subroutine *rdrelv* is detailed in one of the following sections.

Block 30. Subroutine *selrdr_prf* is called to process the radar PRF control.

Block 31. Test whether the desired radar mode, STT target, radar pattern, radar pattern position or PRF control mode is different from the current setting. If true, plant a radar switch change event (block 32).

Block 32. Subroutine *plt_rdr_swch* is called to plant a radar switch change event. The change event is scheduled to occur 0.2 seconds after the decision is made to change the radar. This delay simulates delays due to pilot reaction time and due to the time required for the radar to respond to changes in its switch settings. When the simulation time reaches this event time the radar switch settings will be changed to match desired radar mode.

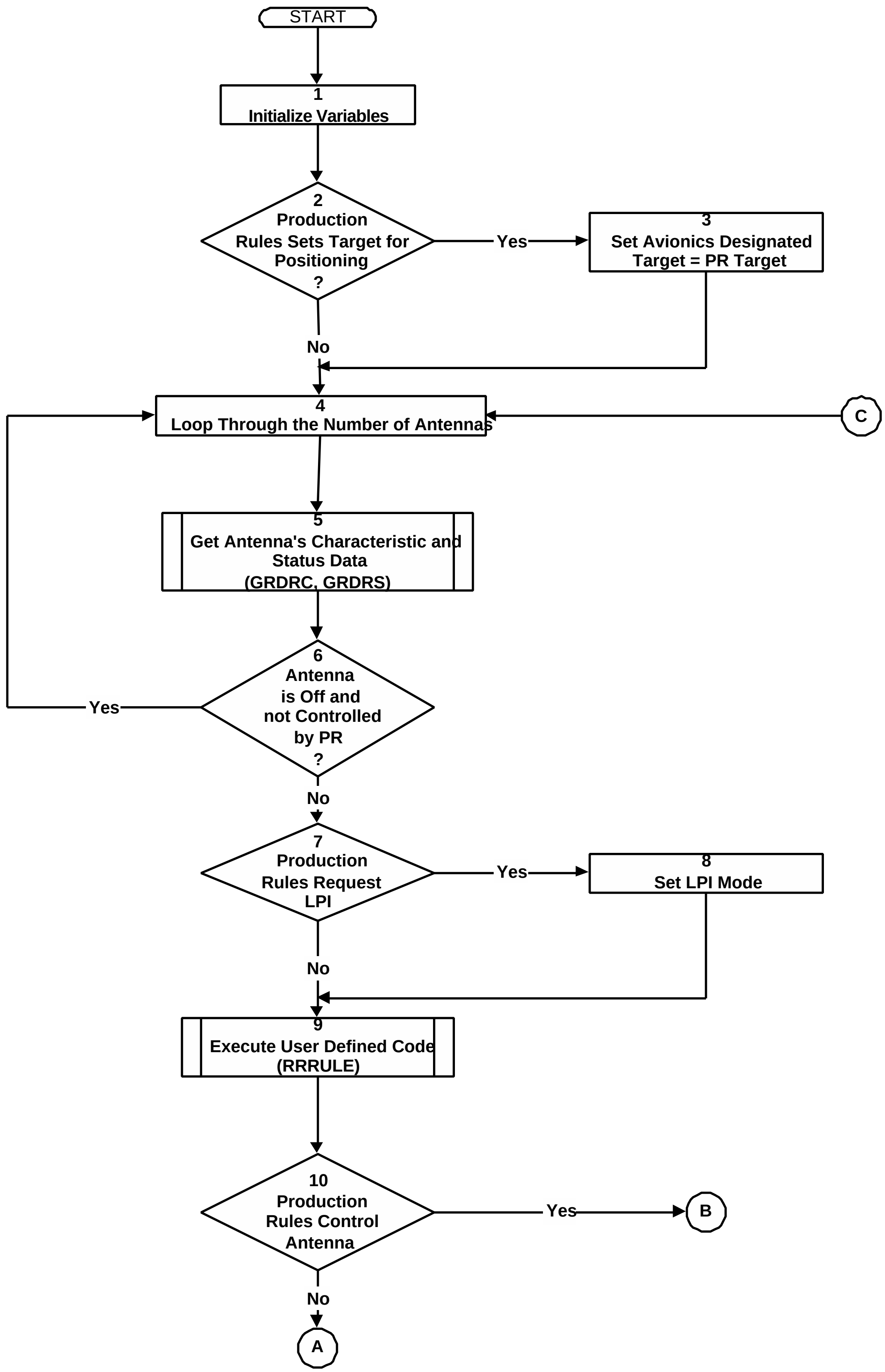

FIGURE 2.22-1. SELRDR Functional Flow Diagram (Page 1 of 3).

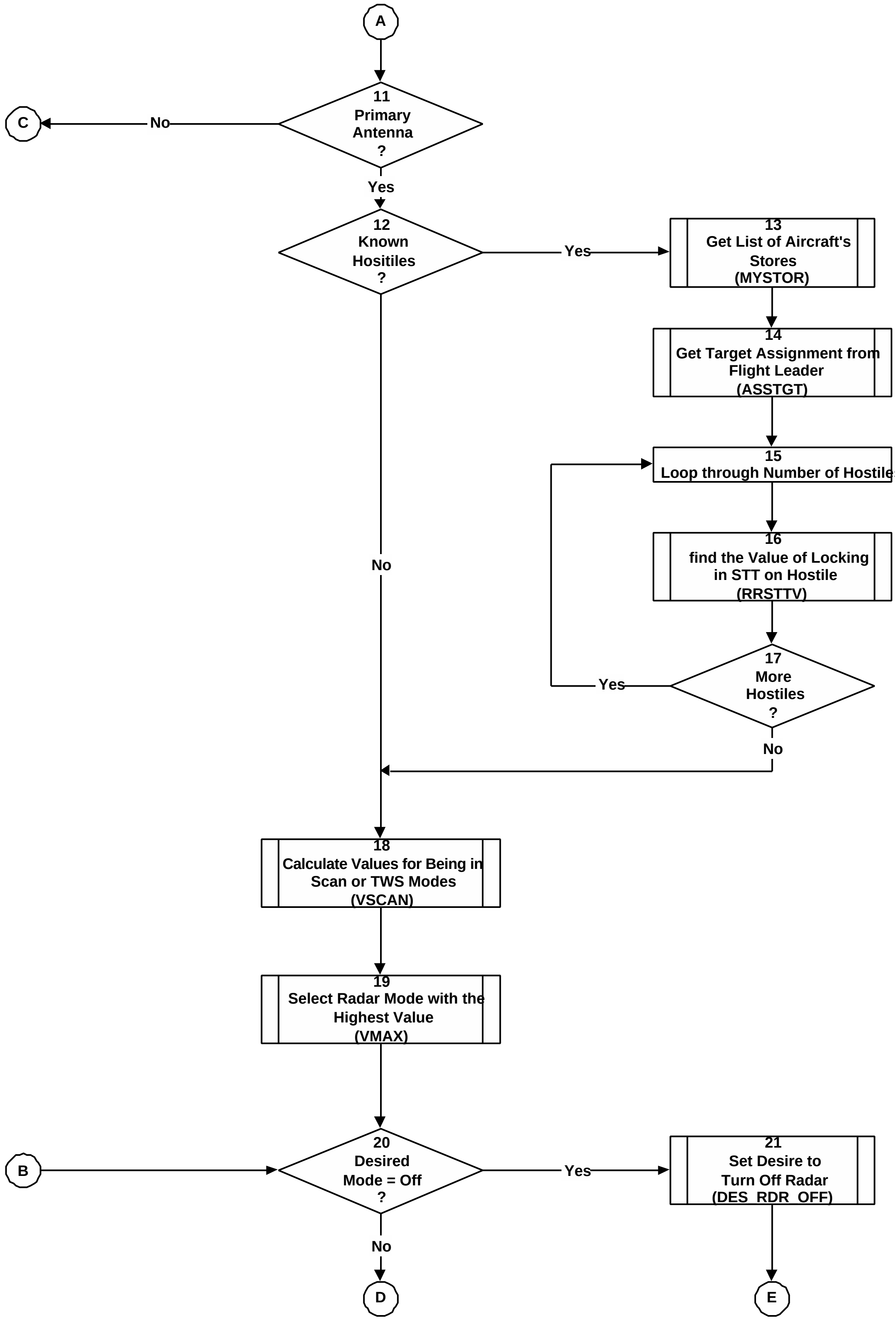

FIGURE 2.22-1. SELRDR Functional Flow Diagram (Page 2 of 3).

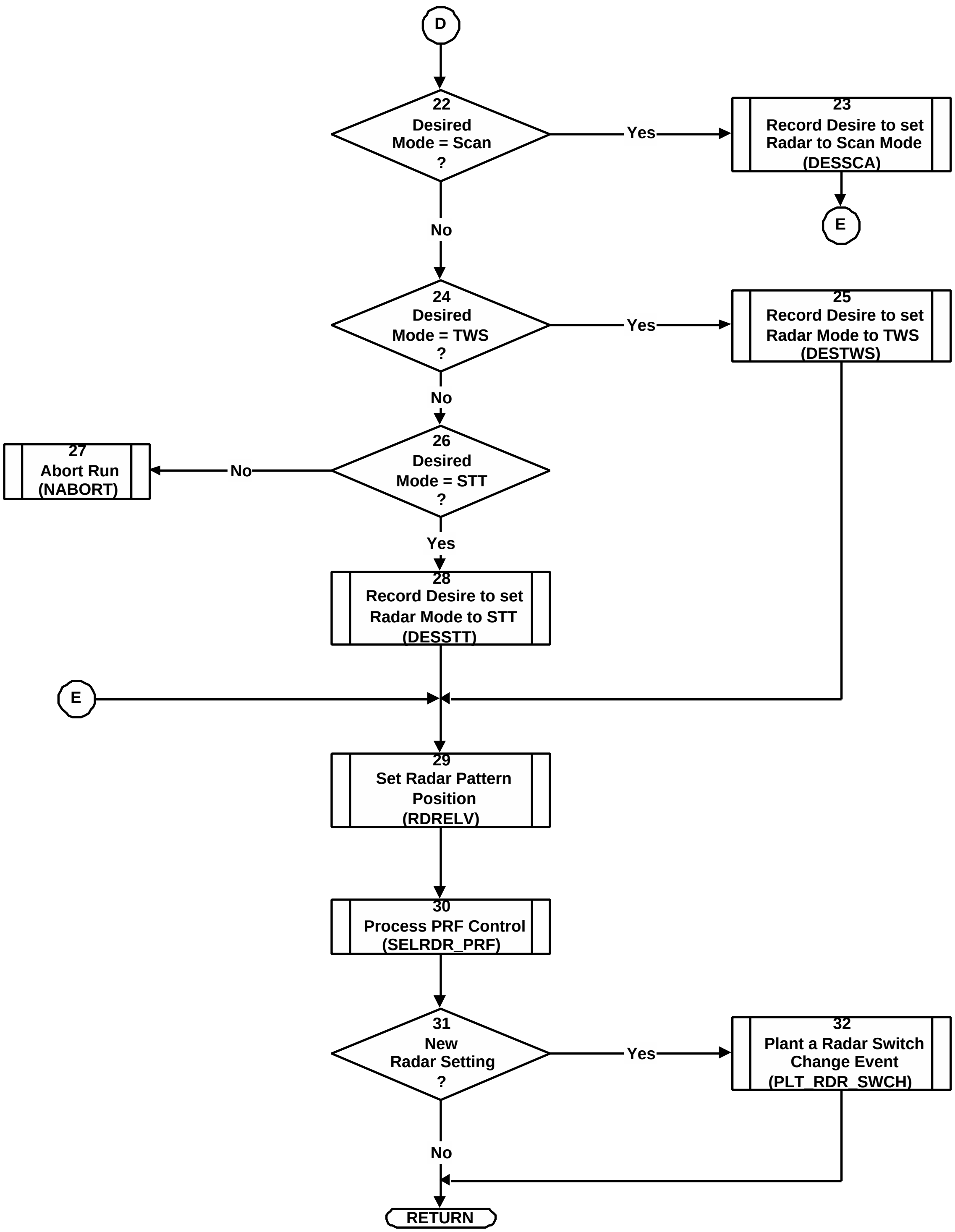

FIGURE 2.22-1. SELRDR Functional Flow Diagram (Page 3 of 3).

Subroutine RRRULE

Subroutine *rrrule* handles production rules control of a radar antenna. Figure 2.22-2 is the functional flow diagram that describes the logic used to implement *rrrule*. The blocks are numbered for ease of reference in the following discussion.

Block 1. Test if production rules control the radar mode selection for this antenna. If false, return to the calling program. Otherwise, continue to Block 2.

Block 2. Test if the current radar mode is STT. If true, continue to Block 3. If not, jump to Block 7.

Block 3. This block represents the top of a DO loop over the number of missiles perceived by the pilot to require illumination by his radar. Note that this is based upon pilot perception and not ground truth.

Block 4. Test if the missile is a semi-active missile and the missile's target is currently locked up in STT mode. If true, go to Block 6. Otherwise, go to Block 5.

Block 5. Test if there are more missiles requiring illumination. If true, process the next missile (go back to Block 4). If not, jump to Block 7.

Block 6. Ignore the production rule directive (leaving the radar in STT) and return to the calling routine.

Block 7. Test if production rules are requesting the use of CIC submode. If true, determine if STT or scan mode has also been specified (Blocks 8-15). If false, jump to Block 16.

Block 8. Test if production rules have specified STT mode as the desired radar mode. If true, continue at Block 9. If false, jump to Block 13.

Block 9. Test if the target is in the conscious pilot's mental model. If it is not, then return control to the calling routine (without setting a desired radar mode), since the pilot cannot be told to lock up on a target if he is not aware of that target.

Block 10. Test if the target is thought to be hostile. If true, set the desired radar mode flag to STT. If false, set the desired radar mode flag to CIC/scan (don't want to lock up on a friendly).

Block 11. Desired radar mode flag is set to CIC/scan, since the target is not a hostile.

Block 12. Desired radar mode flag is set to STT, since the target is a hostile.

Block 13. Test if production rules have specified scan mode as the desired radar mode. If true, continue to Block 14. If false, jump to block 15.

Block 14. Desired radar mode flag is set to scan. Return to calling routine.

Block 15. Subroutine *nabort* is called to abort the run due to an invalid desired radar mode selection in the production rules.

Block 16. Test if production rules has specified OFF as the desired radar mode. If true, continue to Block 17. If false, jump to Block 18.

Block 17. Desired radar mode flag is set to OFF. Return to calling routine.

Block 18. Test if production rules has specified STT as the desired radar mode. If true, continue to Block 19. If false, check for other desired mode selections.

Block 19. Test if the target is in the conscious pilot's mental model. If it is not, then return control to the calling routine (without setting a desired radar mode), since the pilot cannot be directed to lock up on a target unless he is already aware of that target.

Block 20. Test if the target is thought to be hostile. If true, continue to Block 21. If false, then return to the calling routine (without setting a desired radar mode).

Block 21. Desired radar mode flag is set to STT, since the target is a hostile and in the conscious pilot's mental model. Return to the calling routine.

Block 22. Test if production rules has specified scan mode as the desired radar mode. If true, continue to Block 23. If false, jump to Block 24.

Block 23. Desired radar mode flag is set to scan. Return to the calling routine.

Block 24. Test if production rules has selected TWS/SPOT as the desired radar mode. If true, continue to Block 25. If false, an invalid selection has been made, jump to Block 25.

Block 25. Desired radar mode flag is set to TWS. Return to the calling routine.

Block 26. Subroutine *nabort* is called to abort the run due to and invalid radar mode selection in the production rules.

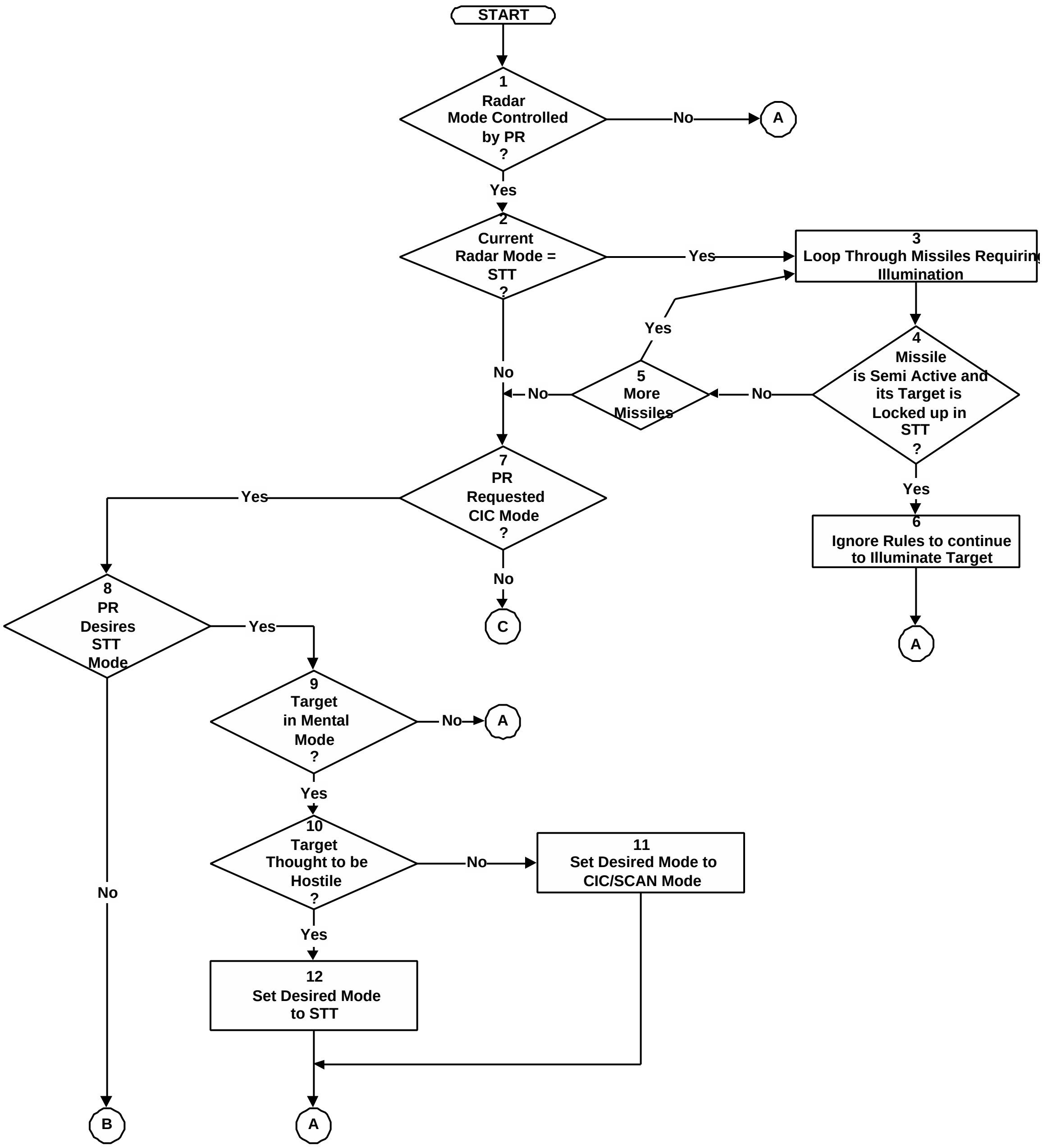

FIGURE 2.22-2. RRRULE Functional Flow Diagram (Page 1 of 2).

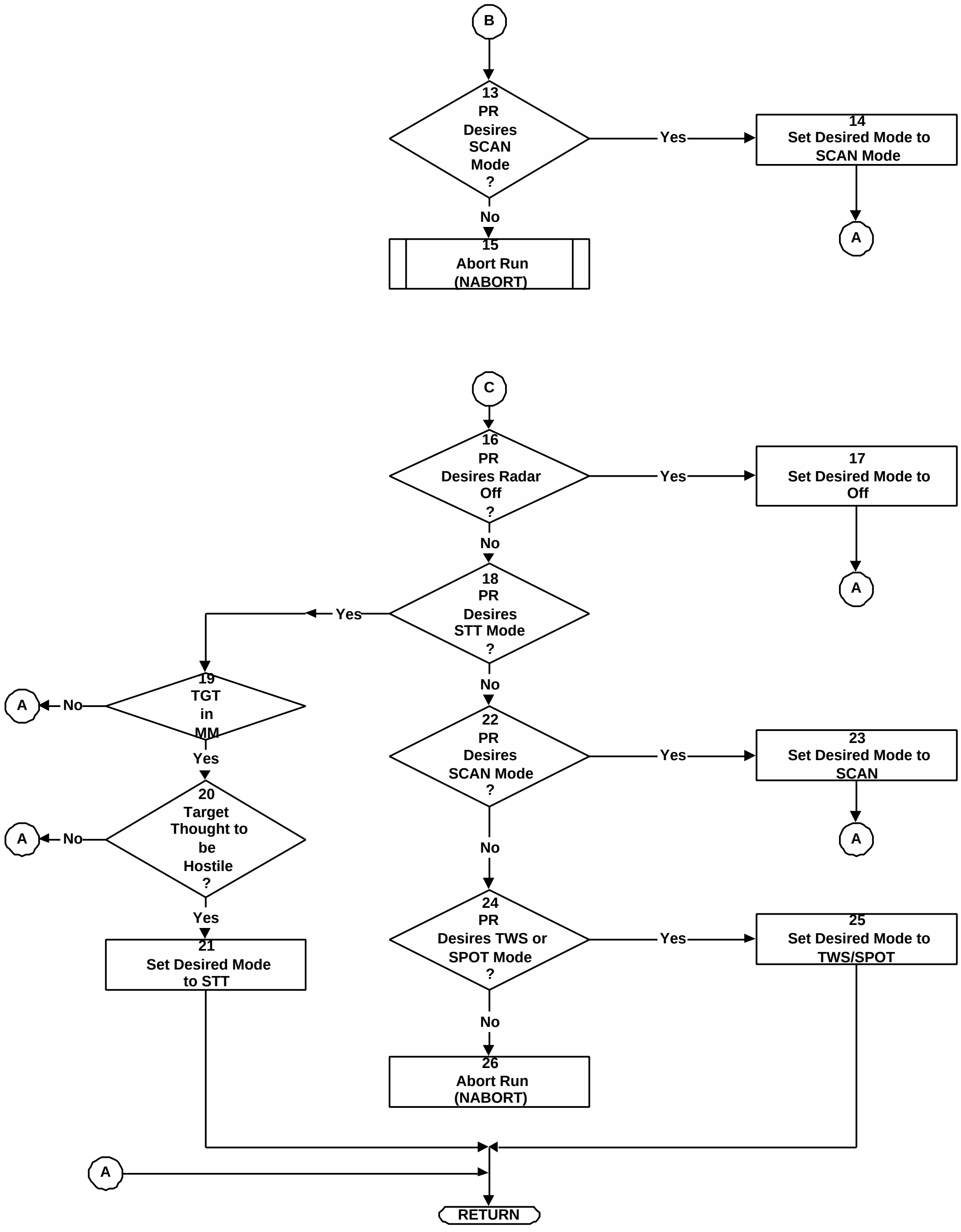

FIGURE 2.22-2. RRRULE Functional Flow Diagram (Page 2 of 2).

Subroutine RRSTTV

Subroutine *rrsttv* computes a value associated with being in STT against a particular aircraft. Early in the routine, a test is made to see if the radar is already in STT, but has no reason to be. (This can happen if the was supporting a missile that has since failed, and the pilot has not selected another weapon requiring STT support.) If so, the value for STT is set to zero and the routine returns. If not, *rrsttv* evaluates a value function for locking up in STT on a specified aircraft. The calculated value is compared in routine *selrdr* to values for being in other radar modes. The mode with the highest score is chosen as the desired radar mode. The value function for locking up in STT consists of 6 components:

$$rdrvlx = vlockd + vslct + vslbvr + vbvr + apvalx + velerr*(vdloc + rvord)$$

where:

vlockd = Value reflecting radar lock status for the target under consideration. If already locked, *vlockd* = 4 if the radar is currently supporting at least one missile, and 4**fmxtgt* otherwise, where *fmxtgt* is the 'maximum target factor' given by
1 - *nmhutl*/*mxtgt_ac*. *nmhutl* is the number of missiles the pilot believes to be currently in the air against the target, and *mxtgt_ac* is the maximum number of missiles allowed by doctrine against that target at any given time, so *fmxtgt* ranges in value from 0 to 1. If not already locked on this target, *vlockd* = 0.

vslct = Value awarded if the hostile under consideration was the target selected for attack in subroutine *selwpn*. If so, *vslct* = 7. If not, *vslct* = 0.

vslbvr = Value awarded if the selected weapon requires support. If *vslct* = 7 and the selected weapon has a semi-active seeker or it will be launched command guided, then *vslbvr* = 3. If not, *vslbvr* = 0. *This extra value is only awarded when vslct is nonzero.*

vbvr = Value awarded if the target under consideration is already the target of a missile requiring semi-active illumination or command guidance from this aircraft. If so, *vbvr* is set to twice the intrinsic value of the target. If not, *vbvr* = 0.

apvalx = A penalty applied to targets outside the radar azimuth and elevation FOR limits. This factor is a combination of border functions reflecting strict and liberal criteria. The penalty starts at 0 for targets within the FOR, drops to -10 for targets just outside the FOR (between 5 and 30 degrees out), then drops again to -15 for targets well outside the FOR (more than 40 degrees out in azimuth and/or elevation).

*velerr**(*vdloc* + *rvord*) =Value associated with trying to lock up. If the radar is already attempting to lock on this target, *vdloc* is set to *fmxtgt*. If not, *vdloc* is set to 0. If the conscious pilot has been ordered to attack the target, *rvord* is set to *fmxtgt*. If not, *rvord* is set to 0. *Velerr* is a multiplier that reflects uncertainty in elevation, so it lowers the value of trying to lock on a target whose position is poorly known.

Figure 2.22-3 is the functional flow diagram that describes the logic used to implement *rrsttv*. The blocks are numbered for ease of reference in the following discussion.

Block 1. Test if the radar antenna being considered is the primary antenna. If not, abort (Block 2) because only primary antennas are currently controlled by *rsttv* (non-primary antennas may only be controlled through production rules). Otherwise, continue processing the STT value function (Block 3).

Block 2. Subroutine *nabort* is called to abort the run due to an attempt to control a non-primary antenna.

Block 3. Current lock status (locked, trying to lock, not locked and not trying to lock) and time to lock are determined. If the current radar mode is STT and the ID of the target aircraft matches the ID of an aircraft that the pilot is locked on, then the locked flag is set equal to 2 (locked on) and the time until lock is set equal to zero. If the antenna is already attempting to lock, and the ID of the target aircraft is the same as the ID of the aircraft that it is trying to lock on, then the locked flag is set equal to 1 (trying for lock) and the time until lock is set to the earliest time to lock minus the current time. Otherwise the locked flag is set to 0 (not locked and not trying to lock) and the time until lock to the average time required to for this antenna to achieve lock.

Block 4. Subroutines *vecinc* and *vsub* are called to compute range and range rate projections from the conscious pilot's aircraft to the target aircraft. Subroutine *vecinc* takes the conscious pilot's position and velocity and projects the position forward in time to the estimated lock time. *Vecinc* is called a second time to project the target aircraft's position to the same time. Subroutine *Vsub* is then called twice, once to find the difference in velocity between the conscious pilot's aircraft and the target aircraft, and a second time to find the difference in position between the two aircraft. Range is calculated as the magnitude of the position difference. Range rate is calculated using the dot product of the velocity difference and the position difference, divided by the projected range.

Block 5. This block represents the top of a DO loop over all of the missiles perceived by the pilot as requiring support from his aircraft at the current time.

Block 6. Test if the hostile under consideration is the target of the missile. If true, go to Block 7. If false, process the next missile requiring support.

Block 7. Set the support flag to true.

Block 8. Test if the missile is a semi-active missile. If true, continue at block 9. If false, go to Block 10.

Block 9. Set semi-active flag to true.

Block 10. Test if there are more missiles requiring illumination to be considered in this loop. If true, return to the top of the loop (block 5).

Block 11. Test whether to bypass scoring for this target. The value function for being in single target track against this target will be set to zero in cases where TWS is available and the radar is already in STT on this target, but there are no semi-active missiles airborne or selected for firing against this target. If this is the case, the value for STT is set to zero and the routine returns.

Block 12. The ‘maximum targeted factor’, *fmxtgt*, is determined. *fmxtgt* is defined above.

Block 13. Test if the current lock designation is equal to one (trying for lock). If true, go to Block 14. If false, go to Block 15.

Block 14. Set *vdloc* = *fmxtgt*.

Block 15. Test if the current lock designation is equal to two (already locked). If true, go to Block 16. If false, go to Block 19.

Block 16. Set *vlockd* = 4.0

Block 17. Test if target under consideration already has missiles fired against it that require support. If not, go to Block 18. If so, jump to Block 19.

Block 18. Multiply *vlockd* by *fmxtgt* to lessen the value of locking up on a target that is already under attack by other weapons.

Block 19. Test if the conscious pilot’s aircraft has any weapons available. If no weapons are available, go to Block 20. If there are weapons, go to Block 21.

Block 20. Set the elevation penalty, *velerr*, to zero. Note that the entire score for STT is not set to zero under this condition, as there may be weapons in the air that still require STT support.

Block 21. Test if the conscious pilot has already locked up on the target aircraft. If true, continue to Block 22. If false, jump to Block 23.

Block 22. Set *velerr* = 1.0, *rtgo* to the current range to the target, and *ttgo* to 0.

Blocks 23-33 calculate the elevation uncertainty penalty when the conscious pilot has weapons and is not already locked up on the target. First the probability of missing the target with the scan volume is calculated, then an engagement time scale is set up and the elevation uncertainty penalty is calculated.

Block 23. The extent of the scan up and down in elevation, *elwid*, is calculated using the number of bars, bar width and the beam spread between half power points.

Block 24. Test the value of the perfect information flag. If true, go to Block 25. If false, go to Block 26.

Block 25. The probability of missing the target with the scan volume is calculated as a Cauchy function, based on the time since last observation and the perceived elevation rate of the target.

Block 26. The probability of missing the target with the scan volume is calculated as a normal function, based on translating the altitude uncertainty into an elevation uncertainty.

The next block of code computes the value if *velerr*. The purpose here is to impose a larger penalty for situations where there is little time available to search for the target while trying to obtain lock. First, an estimate is made of *ttgo*, the time required to achieve lock, based upon the frame time of the radar in scan mode and the probability of missing the target with the scan pattern. Next, an estimate is made of *rtgo*, what the range to the target will be when lock is achieved. Then *tusabl*, the amount of time available to engage after lock, is computed by dividing *rtgo* by the range rate. A weapon range, *scnlmx*, is determined based upon the range of the currently selected weapon or the longest range weapon available if no weapon is currently selected. Finally, the engagement time scale is computed as *scnlmx* divided by 2000 ft/sec.

At this point, we have an estimate of the time that will be available after achieving lock and an estimate of the time taken for our weapon of choice to reach its target if fired near its maximum range. *velerr* is computed using a border function of the difference between *tusabl* and .7**tscale*, with a width of .3**tscale*. The intent is that if lock can be achieved well before the missile needs to be fired, the penalty for trying to achieve lock should be small. If lock probably cannot be achieved in time, the penalty should be increased.

<u>Block 27.</u> The frame time is calculated using the scan volume, scan rate and time delay between bars.

<u>Block 28.</u> *tusabl* is determined by first computing *ttgo* as the minimum of 12 seconds or (2 frame times divided by 1-*pmiss*), where pmiss is the estimated probability that the pilot will miss the target when positioning the scan pattern. Next, *rtgo* is computed as the current range to the target plus the range rate times *ttgo*. Note that range rate is negative for a closing target. Finally, *tusable* is computed as the maximum of *rtgo* and 0.1 divided by the maximum of 0.1 and the negative of the range rate.

<u>Block 29.</u> Test if the selected weapon is a medium range BVR missile. If true, go to Block 30. If false, go to Block 31.

<u>Block 30.</u> Set *scnlmx* to the range of the selected weapon.

<u>Block 31.</u> Set *scnlmx* to the range of the longest range available weapon.

<u>Block 32.</u> The engagement time scale is calculated as *scnlmx* divided by a typical head on closure rate of 2000 ft/sec.

<u>Block 33.</u> The elevation uncertainty penalty is calculated using a smoothed step function (border function). It is driven by the amount of time left until a firing opportunity is present. If a lot of time is available before a firing opportunity is present the penalty approaches its maximum value of one. Once the usable time is less than 70% of the engagement time scale, the penalty function works its way to its minimum value of zero. Since velerr is applied as a multiplier to a positive score, a value of 1.0 means that no penalty is applied and a value of zero means that the maximum penalty is applied.

<u>Block 34.</u> Test if the pilot believes that there are any missiles already in the air against this target that require support. If there are, go to Block 35. If not, go to Block 36.

<u>Block 35.</u> vbvr is set to 2 times the target's intrinsic value.

Block 36. Test if the antenna is in STT mode on this target. If not, go to Block 37. If antenna is in STT, go to Block 39.

Block 37. Normalized coefficients are calculated using a border function based on the distances to the elevation and azimuth gimbal limits. One set of coefficients is computed that becomes large as soon as the target passes outside the azimuth or elevation limits of the antenna FOR. The other set becomes large as the target gets more than 30 degrees outside the FOR limits.

Block 38. The coefficients are combined to determine *apvalx*, the overall penalty for being outside the radar aperture.

Block 39. Test if the conscious pilot has a selected target and his selected weapon is a missile. If true, go to Block 40.

Block 40. Set the launch mode *lnchm* to the actual or desired launch mode of the selected missile, whichever is set.

Block 41. The 'need_stt_flag' variable is set to true if the missile will have a semi-active seeker on at launch or if STT lock is a requirement to fire the missile in the chosen launch mode. Otherwise, it is set to false.

Block 42. The 'need_stt_flag' variable is set to false.

Block 43. Test if the antenna has TWS capability, the target being considered is not the selected target or the 'need_stt_flag' variable is false. If all three conditions are true, go to Block 44. Otherwise, go to Block 46.

Block 44. Test if the missile is a semi-active missile. If not, go to Block 45. If it is, go to Block 52.

Block 45. Set *vbvr* to zero. The reason is that STT is not required and TWS is available. TWS is preferred, since it provides wider situation awareness while also providing the required missile support.

Block 46. Test if the target is the target selected in subroutine *selwpn*. If true, go to Block 47. If not, go to Block 50.

Block 47. Set *vslct* to 7.

Block 48. Test if the weapon will be fired with a semi-active seeker on or if it will be launched in a command guided mode. If either condition is true, go to Block 49. If not, go to Block 50.

Block 49. Set *vslbvr* to 3.

Block 50. Test if the target was assigned by the conscious pilot's flight leader. If true, go to Block 51. If not, go to Block 52.

Block 51. Set *rvord* to *fmxtgt*.

Block 52. The final composite value for STT mode, *rdrvlx*, is computed according to the formula given at the beginning of this section.

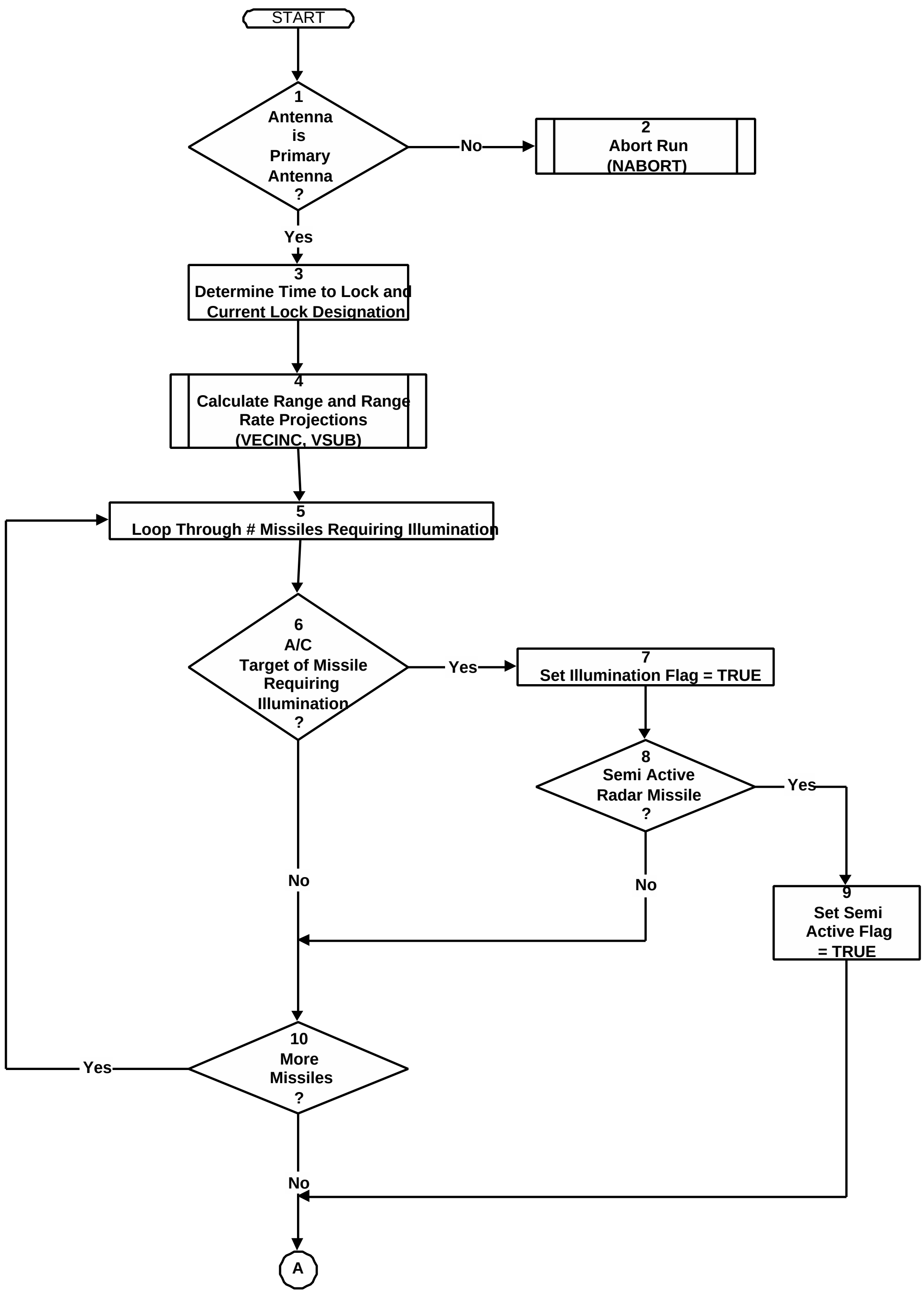

FIGURE 2.22-3. RRSTTV Functional Flow Diagram (Page 1 of 5).

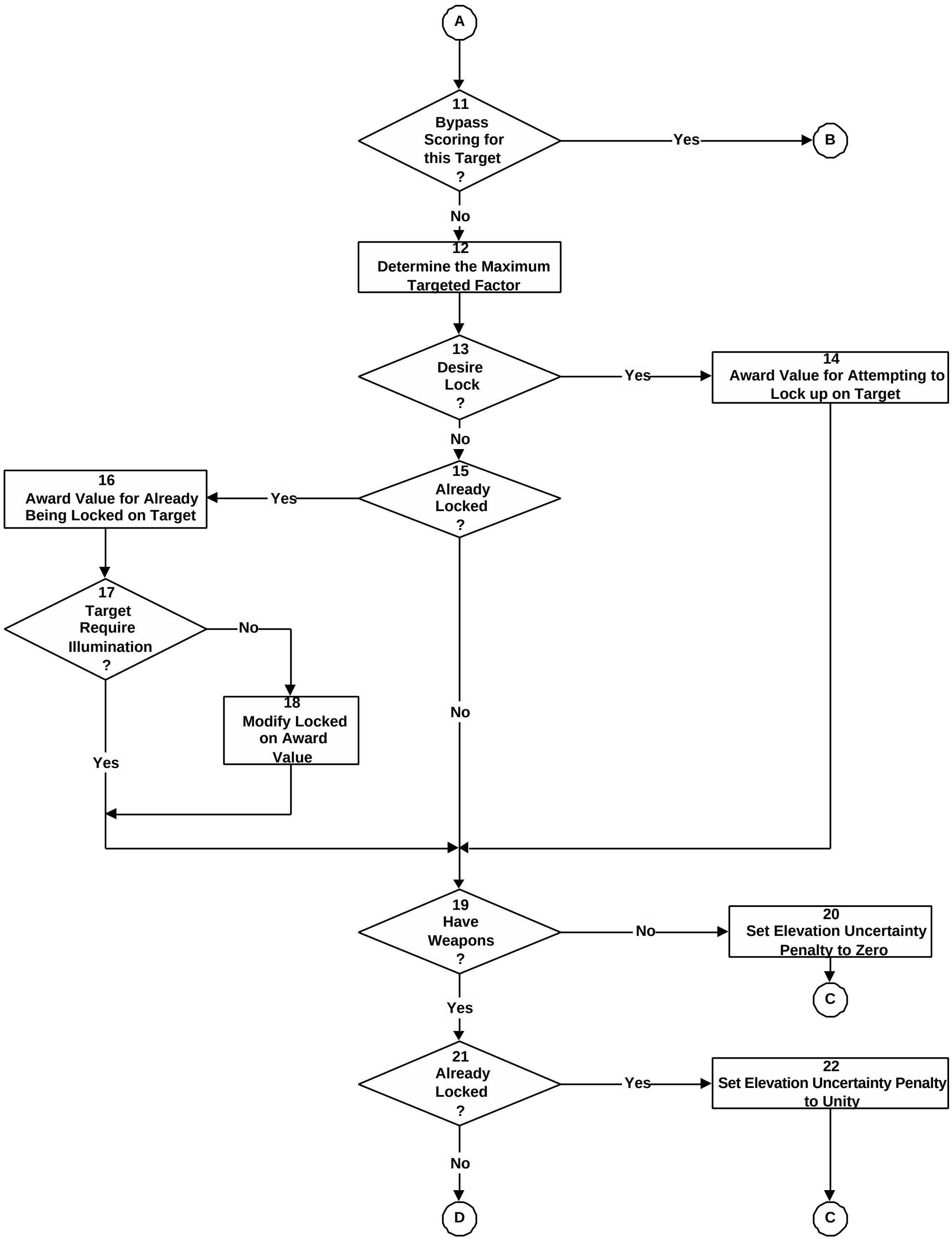

FIGURE 2.22-3. RRSTTV Functional Flow Diagram (Page 2 of 5).

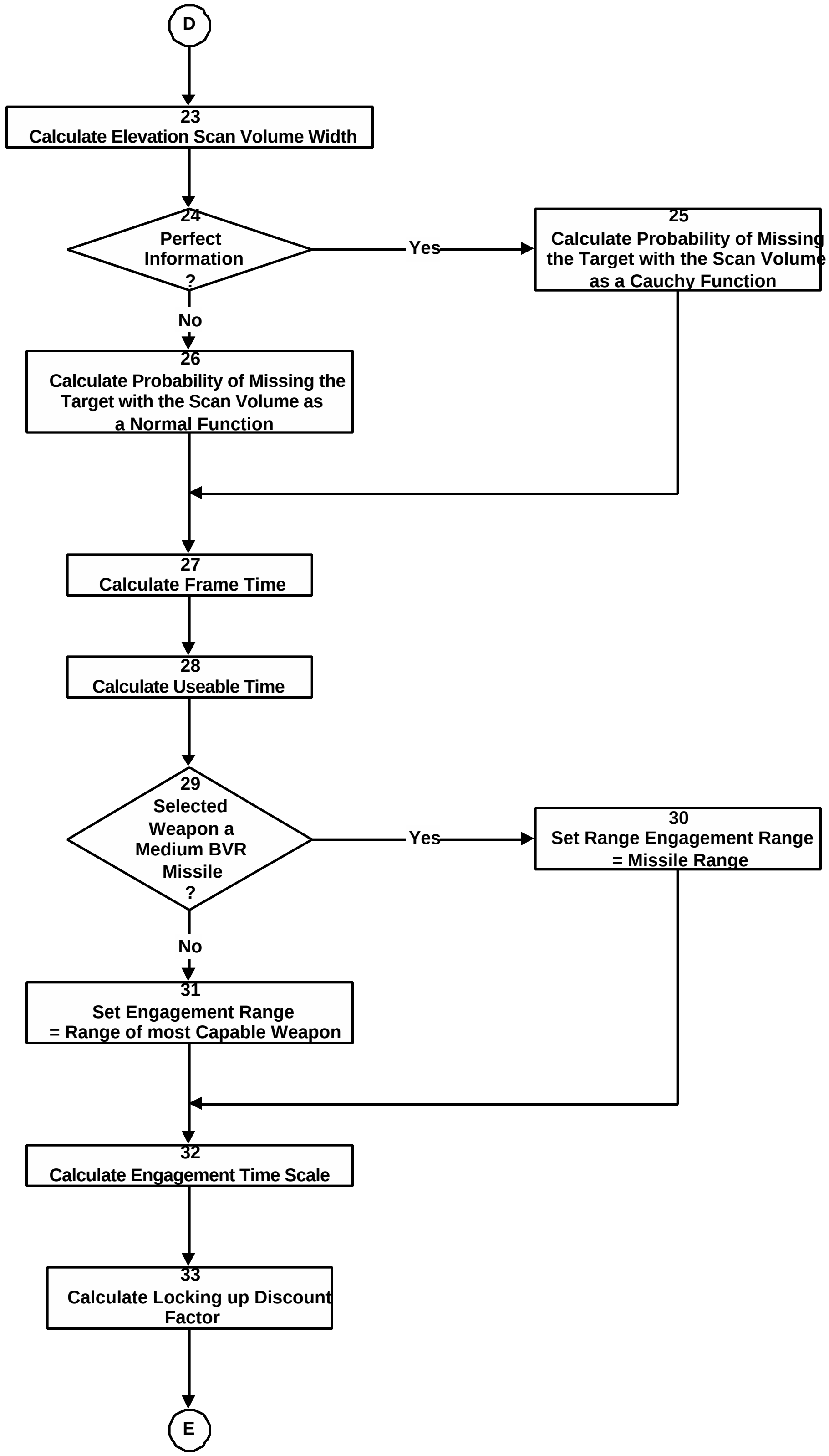

FIGURE 2.22-3. RRSTTV Functional Flow Diagram (Page 3 of 5).

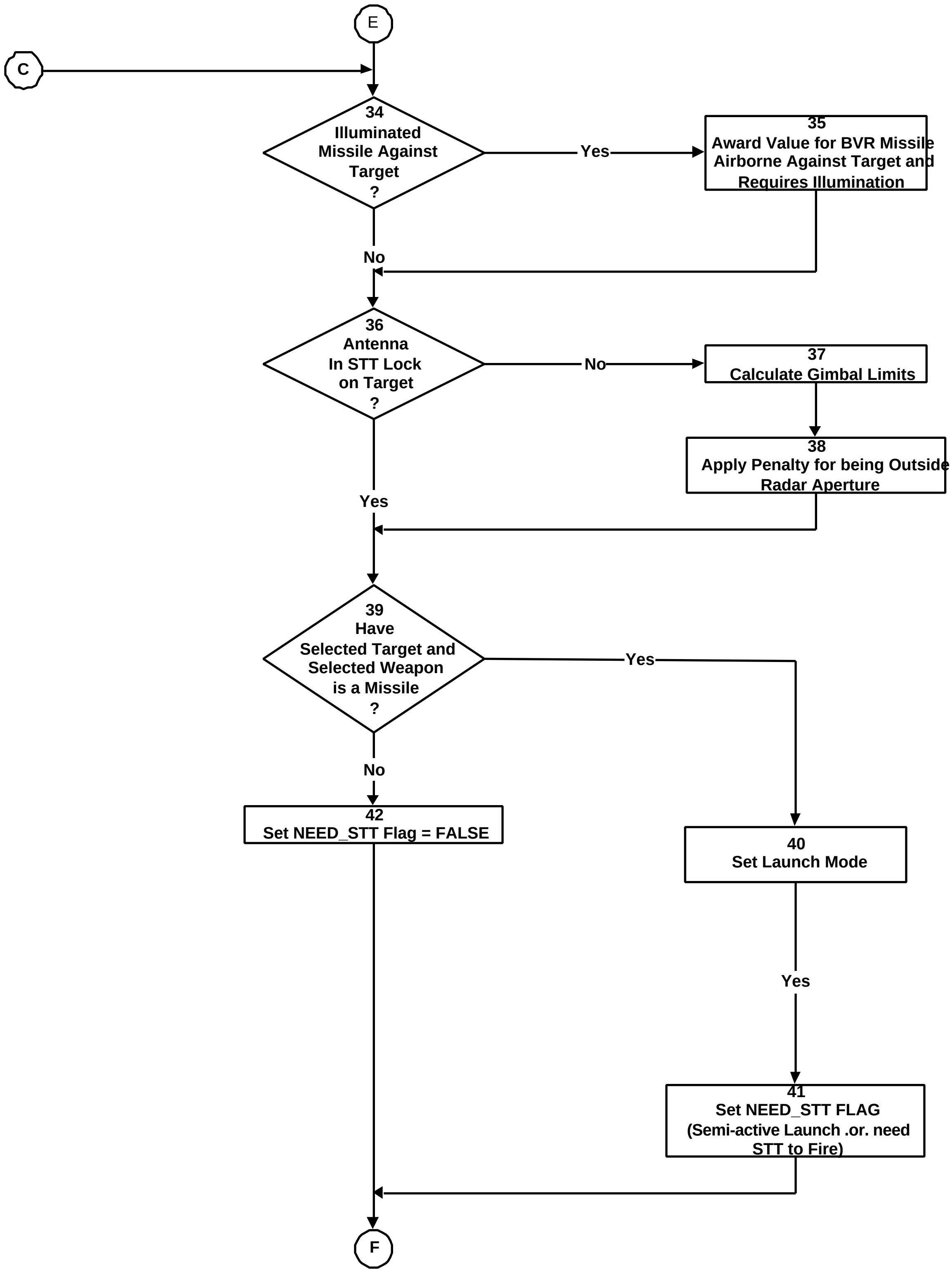

FIGURE 2.22-3. RRSTTV Functional Flow Diagram (Page 4 of 5).

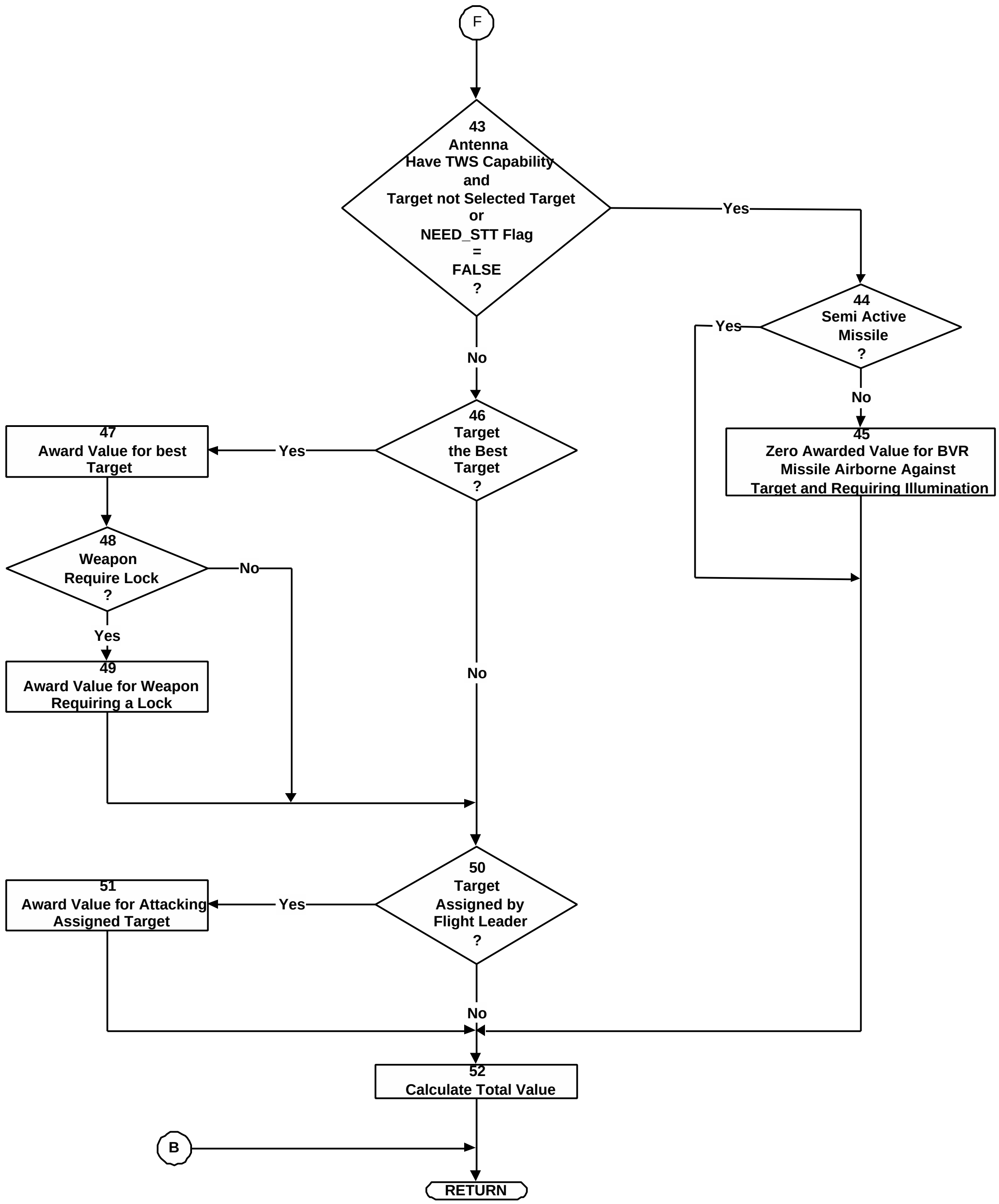

FIGURE 2.22-3. RRSTTV Functional Flow Diagram (Page 5 of 5).

Subroutine VSCAN

Subroutine *vscan* computes scores for being in scan and TWS radar modes. Each calculated value is compared in routine *selrdr* to values for being in other radar modes. The mode with the highest value is selected as the desired radar mode. The value function for being in scan mode consists of 2 components; targets existing within scan range and the relative size of the scan and TWS scan volumes. The value function for being in TWS mode consists of the same two components as the scan value function plus an additional component for existing tracks in the track bank. Figure 2.22-4 is the functional flow diagram that describes the logic used to implement *vscan*. The blocks are numbered for ease of reference in the following discussion.

Blocks 1-9 are used to calculate the first component of the scan and TWS value functions. The first component is based on number of hostiles within the scan range, weapon range and minimum range to hostiles.

<u>Block 1.</u> Test if the pilot is aware of any hostiles. If yes, go to Block 2. If no, go to Block 3.

<u>Block 2.</u> The value of the first component of the scan value function is set equal to a high value (20).

<u>Block 3.</u> Test if the conscious pilot has selected a weapon. If so, go to Block 4. If not, go to Block 5.

<u>Block 4.</u> The range scale, *scnlmx*, is calculated as the maximum of the nominal selection range, *slctr*, for the selected weapon and a range function given by:

*ppmrmx***ppm_rpeak* - *rdot**(*dtloc* + 10)

where:

ppmrmx	=	Max range of the selected weapon.
ppm_rpeak	=	Fraction of max range that defines the “heart” of the envelope.
rdot	=	Range rate between attacker and target. Rdot is negative for a closing geometry.
dtloc	=	Average time required for this antenna to achieve lock.

In other words, this is an estimate of the shortest range that still allows time to lock up the radar and take a good shot.

<u>Block 5.</u> The range scale is calculated as the selection range of the longest range weapon on board. A DO loop encompassing all weapons is executed to find the longest range weapon. If the current weapon in the loop has a range greater than the saved range, the saved range is replaced by the current weapon’s range. When the loop has been completed, the saved range will be the longest available weapon range.

<u>Block 6.</u> The value of the first component of the scan mode value function is calculated using a border function (a smoothed step function ranging from 0 to 1). The value of being in scan will be set to 20 times the border function. The border function will approach a value of 1 when the minimum range to hostiles is greater than the range scale calculated in Block 4 or 5. The border function will approach a value of 0 when minimum range to hostiles is less than the weapon range. The value for the first component of the scan mode value function will be between 0 and 20. The intent of this component is to award a high value to scan mode when all known hostiles are far enough away that there is plenty of time to change to STT, obtain lock, and fire a weapon.

Block 7. Test if the radar antenna has TWS capability. If true, go to Block 8. If false, go to Block 9.

Block 8. The value of the first component of the TWS mode value function is set equal to the value of the first component of the scan mode value function.

Block 9. The value of the first component of the TWS mode value function is set equal to 0, since this antenna is not TWS capable.

Blocks 10-15 are used to calculate the second component of the scan mode and TWS mode value functions. Since the first component of the scan/TWS mode value functions may be the same (block 8), the second component may be the differentiator between the two modes. The second component is based on the scan volume of each mode and is intended to reward radar modes that provide the widest situation awareness.

Block 10. The scan volume for scan mode is calculated from the radar pattern parameters (azimuth halfwidth angle, the bar width, number of bars, etc.).

Block 11. The scan volume for TWS mode is calculated from the radar pattern parameters (azimuth halfwidth angle, the bar width, number of bars, etc.). If the radar does not have TWS capability, the TWS scan volume is set to 0.

Block 12. A determination is made whether TWS or scan has a smaller scan volume. The index *ism* will point to the mode (scan or TWS) with the smaller scan volume. The index *ibig* will point to the radar mode (scan or TWS) with the larger scan volume.

Block 13. The time constant, *trdrin*, for information loss since last the update is calculated. If the antenna is in scan mode, the time constant will be the maximum of 10 seconds or three frames. If it is TWS mode, the time constant will be the sum of 2 + the number of consecutive frames in which detections made are required to establish a track, multiplied by the nominal frame time. This is then used to compute a decay factor given by:

$$decay = \exp(-(time - trdrup)/trdrin)$$

where:

time	=	current simulation time
trdrup	=	time at which this decision was last made for this radar
trdrin	=	time constant calculated above

Block 14. Normalized scores, *vrvol*, are recomputed for transition to scan or TWS in several steps. First, the score for the mode with the smaller volume is reduced by the decay factor to reward transitioning to the mode with the larger volume. Next, if the antenna is currently in STT or off, the scores for both modes are increased by one minus the decay factor to reward transitioning to either mode. Finally, if the antenna is currently in the smaller mode, the value of the larger mode is increased by one minus the decay factor.

Block 15. The value of the second component of the scan mode and TWS mode value functions are calculated based on their scan volumes and the normalized scores computed above. The mode with the larger volume receives a value given by:

*wt**(*vols*(*ism*)**vrvol*(*ism*) + (*vols*(*ibig*) - *vols*(*ism*))**vrvol*(*ibig*))

and the mode with the smaller volume receives a value given by:

*wt***vols*(*ism*)**vrvol*(*ism*)

where:

wt	=	a constant weighing factor for scan or TWS vs. STT, set to 0.16
vols	=	scan volume, indexed by radar mode
vrvol	=	normalized score, indexed by radar mode
ism	=	index of mode with smaller scan volume
ibig	=	index of mode with larger scan volume

The net effect is to give added value to either mode if the radar antenna is off or in STT, and to give additional value to the mode with the larger search volume as a function of how much larger the volume is and how long it has been since the antenna was last in that volume.

Blocks 16-50 are used to calculate the third component of the TWS value function. Value is being in TWS mode for each established aircraft and missile track in the TWS track bank.

<u>Block 16.</u> Test if the radar has TWS capability. If no, go to Block 17. If yes, go to Block 18.

<u>Block 17.</u> *dvtws*, the value of the third component of the TWS value function is set equal to 0.

<u>Block 18.</u> Subroutine *ftkany* is called to retrieve tracks from a radar device.

<u>Block 19.</u> This block represents the top of a DO loop over the number of radar tracks.

<u>Block 20.</u> Test if the radar track is an established track. If true, go to Block 21. If false, get the next radar track.

<u>Block 21.</u> Test if the radar track is an aircraft track. If yes, go to Block 22. If no, go to Block 23.

<u>Block 22.</u> The counter for established tracks is incremented by one.

<u>Block 23.</u> Test if the radar track is a missile track. If yes, go to Block 25. If no, go to Block 24.

<u>Block 24.</u> Subroutine *nabort* is called to abort the run and print diagnostic messages. The run is aborted because the radar model only detects and tracks aircraft and missiles, and this track is neither an aircraft or a missile track, indicating an error.

<u>Block 25.</u> This block represents the top of a DO loop over the number of missiles in the track.

Block 26. Test if the pilot believes that the missile is targeted at him. If yes, go to Block 27. If no, check the next missile in the track.

Block 27. The counter for established tracks is incremented by one.

Block 28. Test if there are more missiles remaining in the track. If true, process the next missile (go back to Block 25). If false, go to Block 29.

Block 29. Test if there are tracks remaining. If yes, go to Block 19 to process the next track. If no, go to Block 30.

Block 30. Initialize variables to be used below. The value for *dvtws*, the third component of the TWS mode value function, is initialized to equal 0.01667 times the lesser of 3 or the number of established tracks. *lnchm*, the launch mode of the selected weapon is set to the actual launch mode if a missile is being fired, the expected launch mode if one has been selected but not yet fired, or zero if no weapon has been selected.

Block 31. Test if the selected weapon is a missile. If yes, go to Block 32. If no, go to Block 37.

Block 32. Subroutine *gmisld* is called to retrieve missile data for the selected missile.

Block 33. Test if TWS should receive additional bias to support an attack. This test is done in two parts. The first part sets the flag *noneed_stt*, which is true if the selected weapon will not have a semi-active seeker on at launch and the intended launch mode does not require STT to fire. The second part sets the flag *bias_tws* to true if noneed_stt is true and if the pilot is in missile mode 1 or 2, indicating that either the pilot has interest in firing or that a shot is imminent. The intent is to bias the pilot toward selecting TWS mode when he is trying to line up a shot that does not explicitly require STT. If bias_tws is true, go to Block 34. If false, go to Block 37.

Block 34. Increase *dvtws* by 7.

Block 35. Test if the selected missile will be fired command guided. If true, go to Block 36. If not, go to Block 37.

Block 36. Increase *dvtws* by 3.

Block 37. Test if the pilot has a selected target. If true, go to Block 38. If not, go to Block 41.

Block 38. Subroutine *lockan* is called to determine if the radar antenna is locked on the target or has an established TWS track. In either case, the variable *lockd* is returned with a value of 2.

Block 39. If lockd = 2, go to Block 40. If not, go to Block 41.

Block 40. Increase *dvtws* by 1.

Block 41. Subroutine *asstgt* is called to retrieve the assigned target from the pilot's flight leader.

Block 42. Test if the pilot has been assigned a target. This is actually a two-part test on variables *aidord* and *virtl*. *aidord* is the tail number of the assigned target, and is nonzero if an assignment has been made. *virtl* is a logical flag that indicates that this is a "virtual" target assignment, which means that the conscious pilot has been ordered to cover another aircraft that was ordered to make the actual attack on *aidord*. If *aidord* is nonzero and *virtl* is false, go to Block 43. Otherwise, go to Block 44.

Block 43. Increase *dvtws* by 1.

This next block of code adds value to TWS to reward supporting command guided missiles that have not yet acquired their targets. An incremental value is computed for each target under attack by this aircraft. This begins at the intrinsic value of the target, but is reduced by one-half for each already acquired missile that is currently airborne against it. Then, for each command guided missile being supported by this aircraft, the value of its target is added to *dvtws*. If the target is under attack by more than one command guided missile, the amount added to *dvtws* is reduced by one-half for each successive missile.

Block 44. Test if there are any command guided missiles from this aircraft currently in the air. If so, go to Block 45. If not, go to Block 51.

Block 45. An incremental target value is created for each hostile to determine how much additional value should be awarded to the TWS mode value function for each unacquired command guided missile targeted at it. The target value is initialized to the target's intrinsic value.

Block 46. The intermediate target value is reduced by 50% if another missile has already acquired the target. A DO loop over each acquired missile is executed to reduce the target value for that target. The reasoning here is that the presence of autonomous missiles against this target lessens the need to also support command guided missiles.

Block 47. This block represents the top of a DO loop over the number of BVR missiles.

Block 48. An additional value equal to half the target value is added *dvtws* if the missile is command guided.

Block 49. The target value for the missile's target is reduced by 50%, in case there is another unacquired BVR missile on the list targeted against it.

Block 50. Test if there are more unacquired command guided missiles on the list. If true, return to the top of the loop (block 47) to process the next missile.

Block 51. The composite value for being in scan mode is calculated by adding the *vlscan* and *dvinfo(scan)* components computed above. The composite value for being in TWS mode is calculated by adding the *valtws*, *dvinfo*(*tws*), and *dvtws* components computed above.

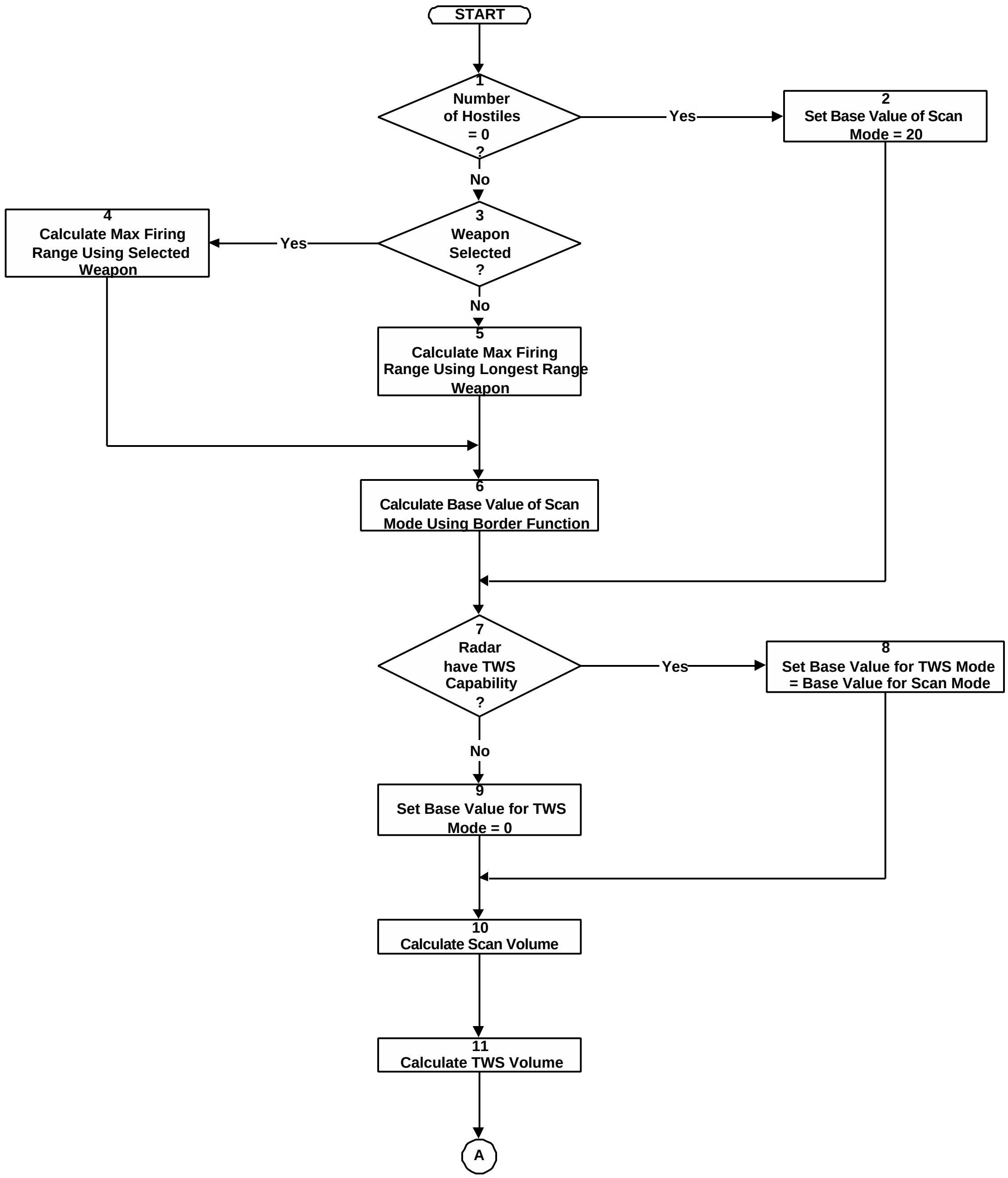

FIGURE 2.22-4. VSCAN Functional Flow Diagram (Page 1 of 5).

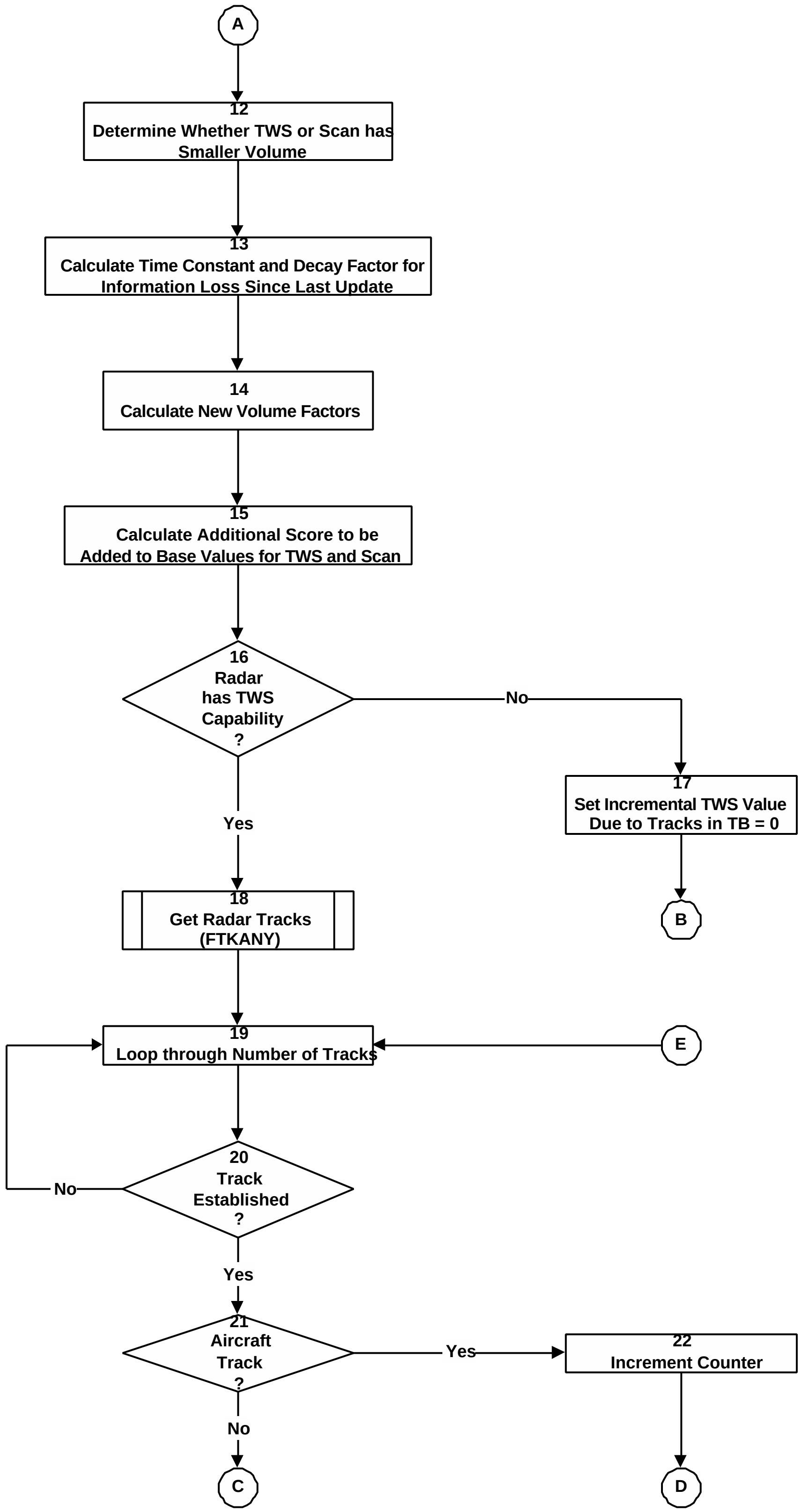

FIGURE 2.22-4. VSCAN Functional Flow Diagram (Page 2 of 5).

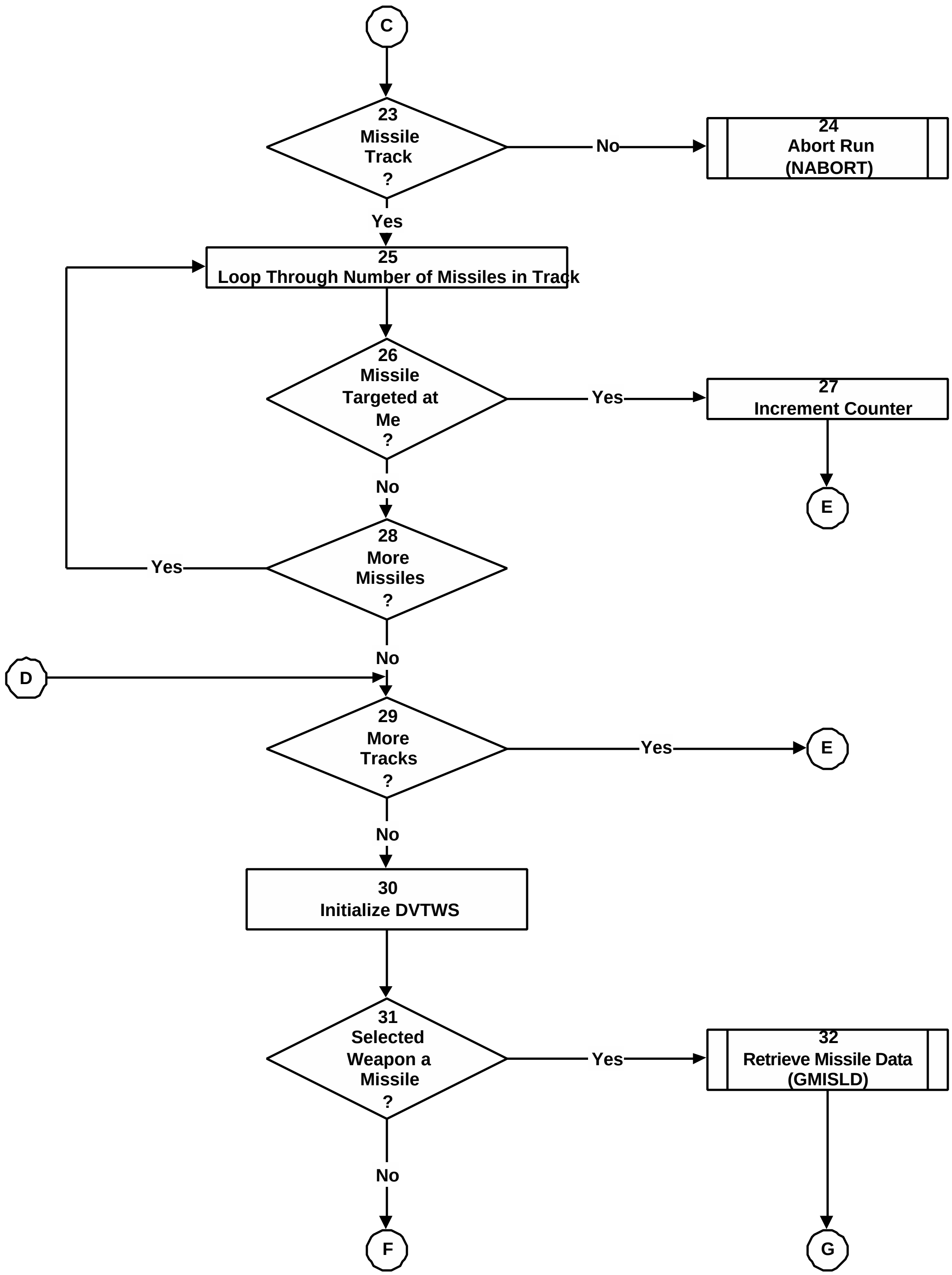

FIGURE 2.22-4. VSCAN Functional Flow Diagram (Page 3 of 5).

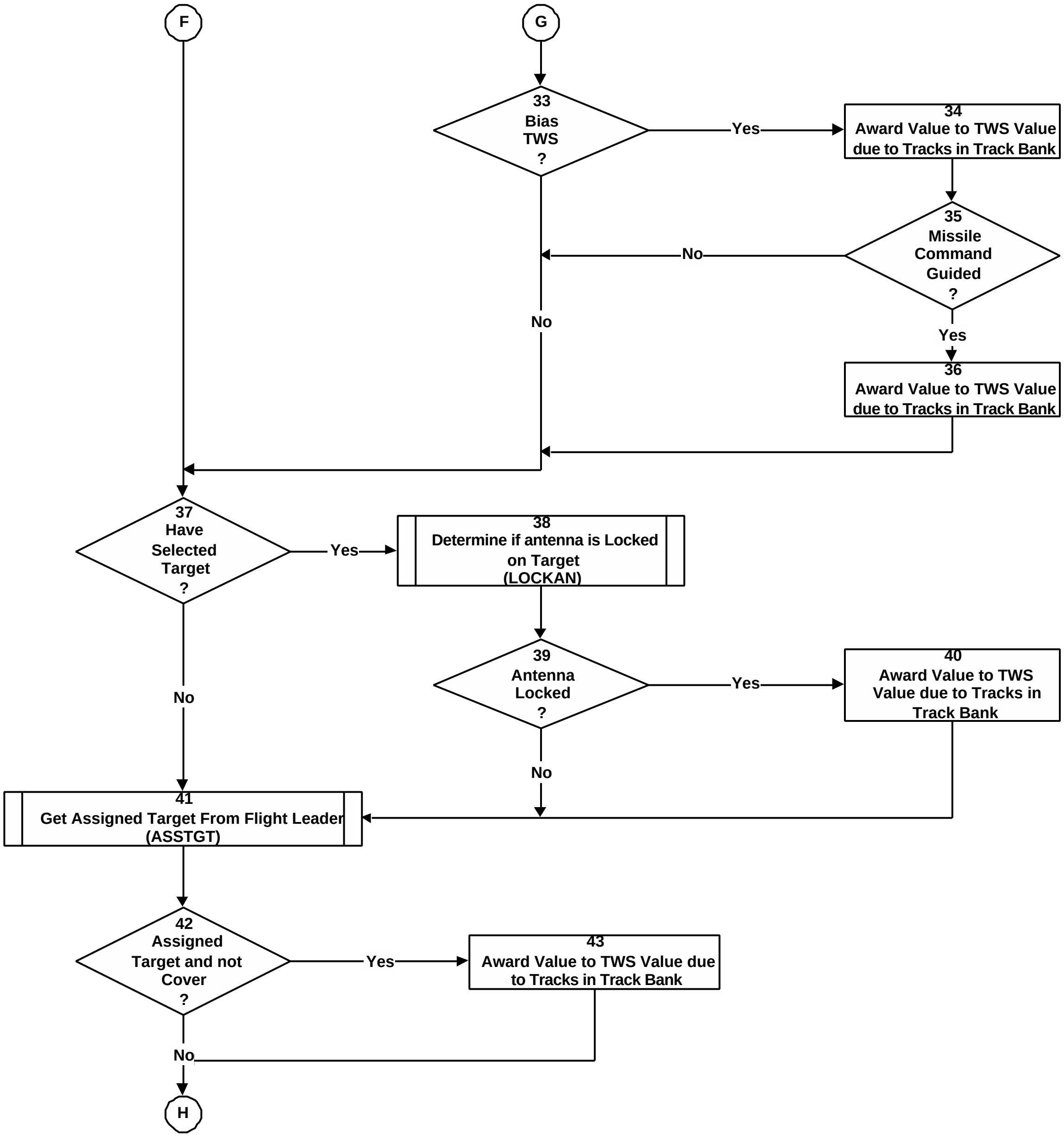

FIGURE 2.22-4. VSCAN Functional Flow Diagram (Page 4 of 5).

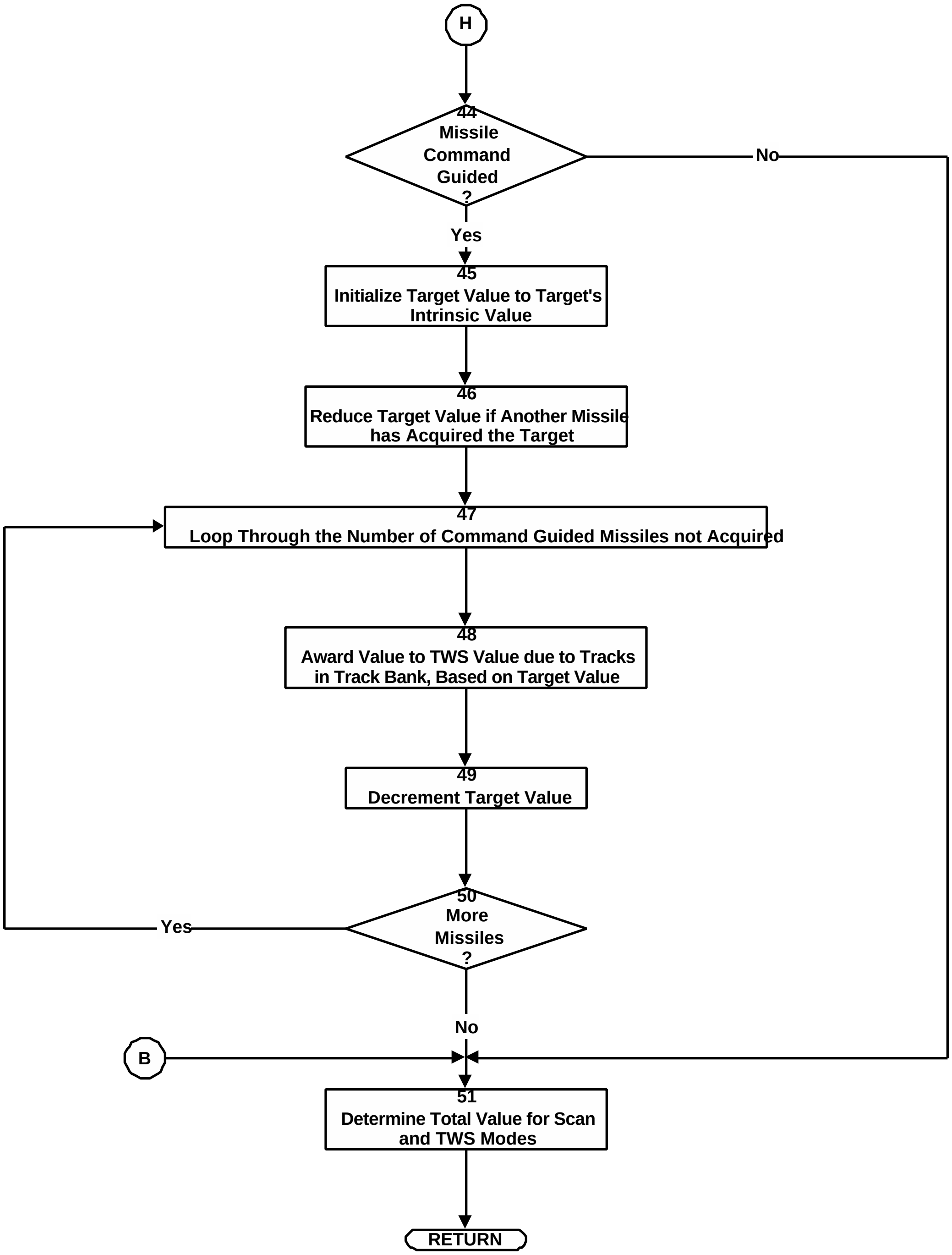

FIGURE 2.22-4. VSCAN Functional Flow Diagram (Page 5 of 5).

Subroutine DESSCA

Subroutine *dessca* records the desire for the radar to be in scan mode. Figure 2.22-5 is the functional flow diagram that describes the logic used to implement *dessca*. The blocks are numbered for ease of reference in the following discussion.

Block 1 Variables that have to do with desired radar mode are initialized, they will have values calculated later in the routine. The radar characteristics and status data are retrieved from memory for this antenna by calling subroutines *grdrc* and *grdrs*.

Block 2 Test if the production rules control the radar mode. If true, go to Block 3. If false, go to Block 7.

Block 3 Test if the radar mode desired by the production rules is scan mode. If true, go to Block 5. If false, go to Block 4.

Block 4 Subroutine *nabort* is called to abort the run and print diagnostic messages. This situation should never occur and would indicate an error condition.

Block 5 Flags are set to indicate whether changes in desired radar mode and submode have been requested. This is done by comparing the desired values recorded from the previous decision with the values desired now. Changes will be recorded in desired mode and in any of the submodes; CIC, burst, or velocity search.

Block 6 Once changes in desired mode and submode have been recorded, the variables that hold desired values are reset to record the new values.

Block 7 Flags are set to indicate changes in the desired mode and submode by the pilot model. This is done by comparing the desired values recorded from the previous decision with the values desired now. Changes will be recorded in desired mode and in any of the submodes; CIC, burst, or velocity search.

Block 8 Once changes in desired mode and submode have been recorded, the variables that hold desired values are reset to record the new values. Note that all submodes are set to false, since the default logic does not request these. They must be explicitly requested via production rules.

Block 9 Test if production rules control the antenna pattern. If yes, go to Block 10. If no, go to Block16.

Block 10 Checks are made to see if the pattern center has changed in azimuth or elevation. A significant change is declared if the difference between the old and new values is larger in magnitude than 0.01 times the average of the old and new values. This test is applied separately to the azimuth and elevation values.

Block 11 Once changes have been tested for, record the new desired azimuth value.

Block 12 Once changes have been tested for, record the new desired elevation value.

Block 13 Check for changes in either the desired number of bars or the desired pattern azimuth halfwidth.

Block 14 Record the new desired number of bars.

Block 15 Record the new pattern azimuth halfwidth.

Block 16 Test if the desired radar mode has changed. If yes, go to Block 18. If no, go to Block 17.

Block 17 Record in the pilot's mental model, that there are no desired changes to the radar pattern or radar pattern position (since the desired radar mode has not changed).

Block 18 Set flags indicating a new desired pattern and new desired pattern position.

Block 19 Test if the CIC submode was requested by production rules (see block 5 above). If yes, go to Block 21. If no, go to Block 20.

Block 20 CIC mode was not desired, so set the desired number of bars, azimuth half width, azimuth of pattern centroid and elevation of pattern centroid to the default values for scan mode. These values were read in from the **SCNRIO** file during initialization.

Block 21 CIC mode is desired, so set the desired number of bars, azimuth half width, azimuth of pattern centroid and elevation of pattern centroid to the default values for CIC mode. These values were read in during initialization as part of the radar's characteristics data.

Block 22 Test if a new desired pattern position was chosen. If yes, go to Block 23. If no, go to Block 24.

Block 23 Subroutine *vpoint* is called to convert the desired pattern centroid azimuth and elevation to a LOS vector in radar heading coordinates. Radar heading coordinates is a coordinate system with it X axis pointed out the aircraft's nose and its Y axis parallel to the ground and pointed to the right. Subroutine *getrhe* is called to calculate the earth to heading rotation matrix. Subroutine *vxfrmc* is then called to transform the LOS vector from heading coordinates to earth coordinates, this is the desired pattern direction, *antdird*.

Block 24 Test if a new radar pattern position is desired and if the production rules are not controlling the radar pattern. If yes, go to Block 25. If no, return.

Block 25 Reset *tmkvsh*, the time that the scan pattern was last shifted, to the current simulation time.

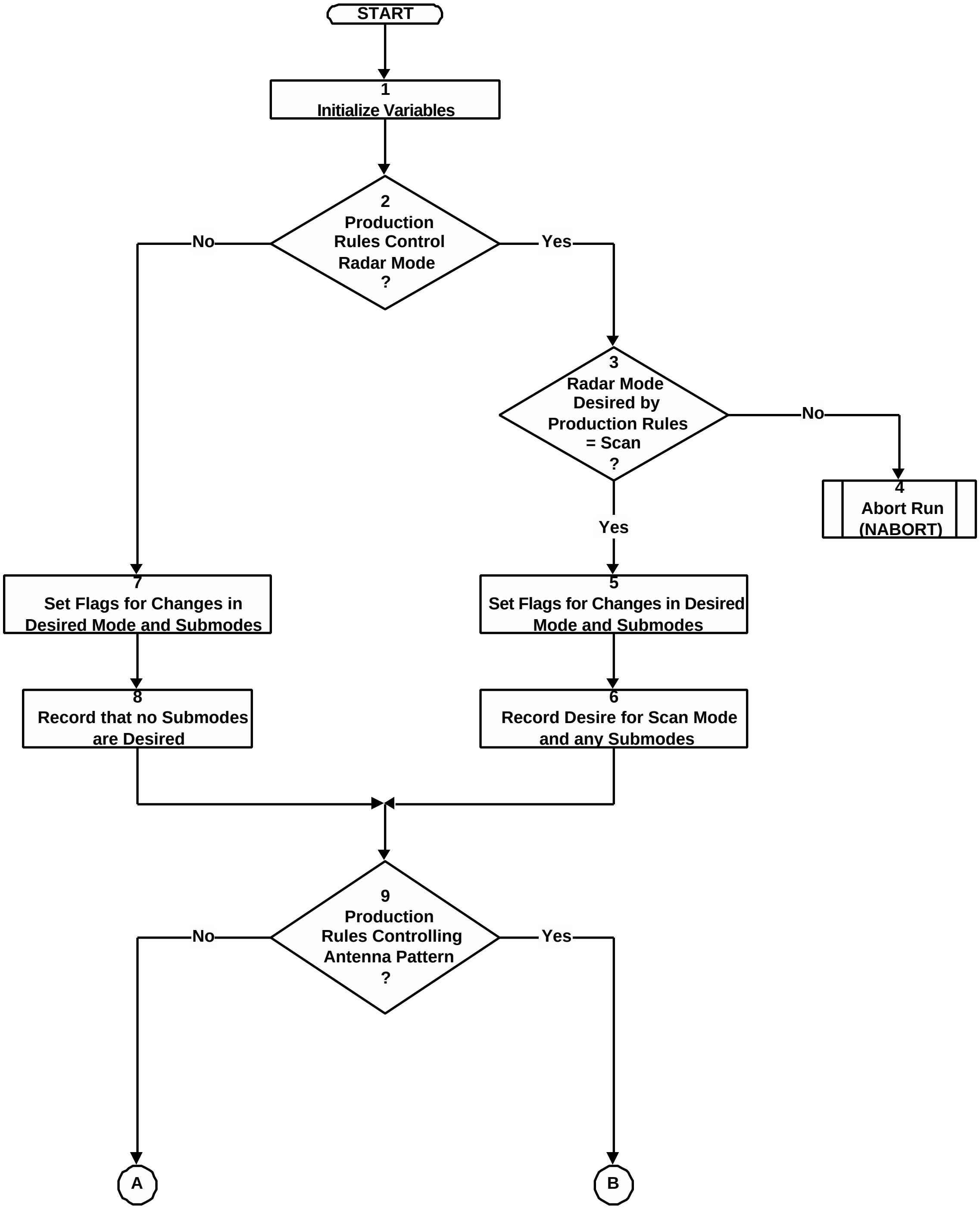

FIGURE 2.22-5. DESSCA Functional Flow Diagram (Page 1 of 3).

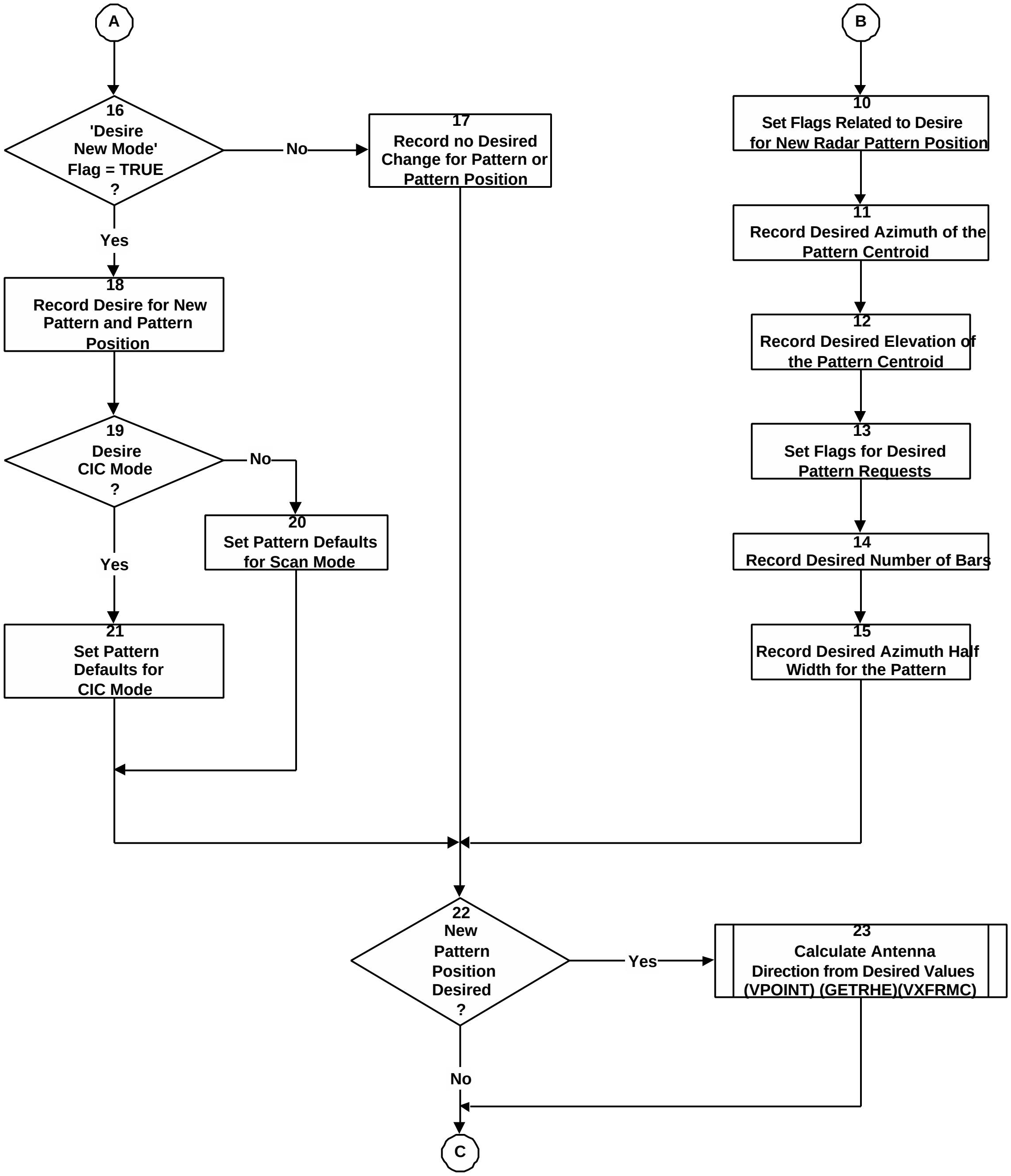

FIGURE 2.22-5. DESSCA Functional Flow Diagram (Page 2 of 3).

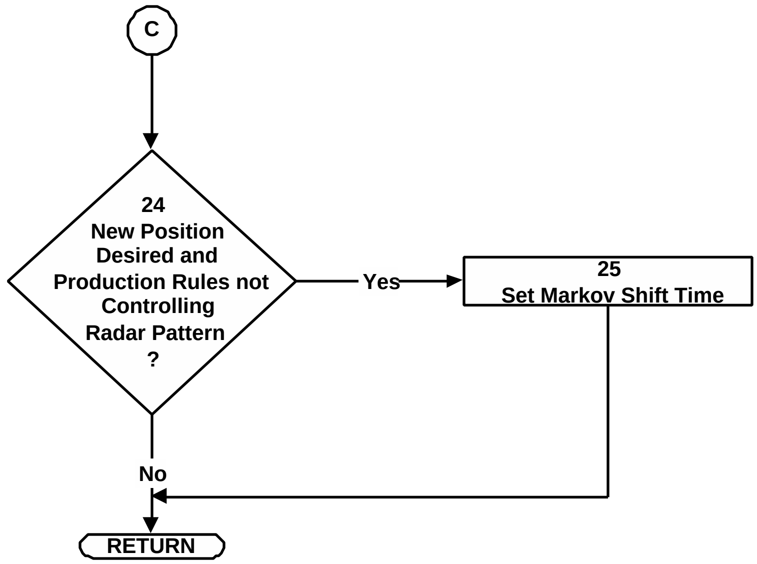

FIGURE 2.22-5. DESSCA Functional Flow Diagram (Page 3 of 3).

Subroutine DESTWS

Subroutine *destws* records the desire for the radar to be in TWS mode. Figure 2.22-6 is the functional flow diagram that describes the logic used to implement *destws*. The blocks are numbered for ease of reference in the following discussion.

<u>Block 1.</u> Variables that have to do with desired radar mode are initialized, they will have values calculated later in the routine. The radar characteristics and status data are retrieved from memory for this antenna by calling subroutines *grdrc* and *grdrs*.

<u>Block 2.</u> Test if the radar antenna is TWS capable. If true, continue at Block 4. If false, go to Block 3.

<u>Block 3.</u> Subroutine *nabort* is called to abort the run and print diagnostic messages. This situation occurs if the desired radar mode is TWS and the radar antenna is not TWS capable. This is purely a defensive check, as this situation should never occur during normal operation.

<u>Block 4.</u> Test if the production rules control the radar mode. If true, continue at Block 5. If false, go to Block 9.

<u>Block 5.</u> Test if the radar mode desired by the production rules is TWS mode. If true, jump to Block 7. If false, go to Block 6.

<u>Block 6.</u> Subroutine *nabort* is called to abort the run and print diagnostic messages. This situation occurs when the production rules desire a radar mode other than TWS (this routine should only be called when TWS mode is desired).

<u>Block 7.</u> Flags are set to indicate changes in desired radar mode and submode. This is done by comparing the previously desired values with the values desired now. Changes will be detected in the desired mode and in the CIC, burst, and velocity search submodes.

If there are changes in the desired values of any of the submodes, the ‘desire new submode’ flag is also set to true.

Block 8. Record the desired values for radar submodes requested by production rules.

Block 9. Flags are set to indicate changes in desired radar mode and submode. This is done by comparing the previously desired values with the values desired now. The ‘desired new mode’ flag is set to true if the previously desired radar mode is not equal to TWS. Changes in the desired values of the submodes are detected by recording the previously desired values for the submodes. This works because the default pilot model will not request any submodes, resulting in new desired values of ‘false’ for all submodes. If any submodes were true last time, then this will represent a change. If they were false, then there will be no change in their desired settings. If there are changes in the desired values of any of the submodes, the ‘desire new submode’ flag is also set to true.

Block 10. Record the desired settings for radar submodes requested by the pilot model. The settings for desired CIC, burst and velocity search submodes are all set to ‘false’.

Block 11. Record TWS as the desired radar mode. Recording this will cause the radar switches settings to be changed elsewhere in Brawler after an appropriate delay has elapsed, simulating the time for the pilot to manipulate the actual switches on the cockpit radar instrumentation.

Block 12. Test if the production rules control the radar antenna pattern. If yes, go to Block 13. If no, go to Block 19.

Block 13. The ‘desire new azimuth center’ flag is set to true if the radar pattern azimuth center selected by production rules is not equal, within a tolerance, to the current radar pattern azimuth center. The ‘desire new elevation center’ flag is set to true if the radar pattern elevation center requested by production rules is not equal, within a tolerance, to the current radar pattern elevation center. If either of these two pattern position flags is set to true, the ‘desire new pattern position’ flag is also to true. The tolerance test used here is that a significant change is declared if the difference between the old and new desired settings differs by more than .001 times the average of the old and new values.

Block 14. Record the desired pattern azimuth centroid specified by production rules.

Block 15. Record the desired pattern elevation centroid specified by production rules.

Block 16. Check for changes in the desired number of bars and azimuth halfwidth of the TWS pattern. The same fractional tolerance test is used for azimuth halfwidth as was used above for pattern positioning. If either of these two pattern flags is set to true, the ‘desire new pattern’ flag is also set to true.

Block 17. Record the desired number of pattern scan bars.

Block 18. Record the desired pattern azimuth half width.

Block 19. Test if a change in the desired radar mode or submode was just detected above (‘desired new mode’ or ‘desire new submode’ flags set above). If true, go to Block 20. If false, go to Block 23.

Block 20. Set flags for new desired pattern and new desired pattern position to true, since a change in mode implies a change in pattern and pattern position.

Block 21. Call subroutine *twspat* to chose the best TWS pattern.

Block 22. Record the desired azimuth half width and number of bars determined from the best TWS pattern. Also record the default pattern centroid azimuth and elevation.

Block 23. Set the flag for new desired pattern position to false.

Block 24. Subroutine *twspat* is called to chose the best TWS pattern.

Block 25. Set the flag for new ‘desired pattern’ to true if the best TWS pattern just selected is not the same as the current TWS pattern.

Block 26. Record the desired azimuth half width and number of bars determined from the best TWS pattern. Also record the pattern centroid azimuth and elevation to be the same as the current pattern centroid azimuth and elevation.

Block 27. Test if a new desired pattern position was chosen. If true, go to Block 28. If false, go to Block 29.

Block 28. Subroutine *vpoint* is called to convert the desired radar pattern centroid azimuth and elevation to a LOS vector. Subroutine *getrhe* is called to calculate the earth to heading rotation matrix. Subroutine *vxfrmc* is then called to transform the LOS vector from heading coordinates to earth coordinates, this is recorded as the desired antenna direction.

Blocks 29-35 perform a check to make sure that the TWS pattern chosen is valid.

Block 29. This block represents the top of a DO loop over the number of valid TWS patterns.

Block 30. Test if the desired TWS pattern matches this TWS pattern. If true, set a variable indicating that a match was found and which pattern was matched. If false, continue checking against valid TWS patterns.

Block 31. Test if more valid TWS patterns remain to be tested against. If true, check the next valid TWS pattern with the desired TWS pattern (go back to Block 29). If false, go to Block 32.

Block 32. Subroutine *nabort* is called to abort the run and print diagnostic messages. This situation occurs when all the valid TWS patterns have been tested against the desired TWS pattern and no match was found.

Block 33. The desired TWS pattern matched a valid TWS pattern, so save the pattern number of the matched pattern.

Block 34. Test if production rules control the radar pattern and if the desired production rules pattern does not match any valid TWS patterns. If true, abort the run. If false, record the desired TWS pattern in the pilot's mental model.

Block 35. Record the desired TWS pattern.

Block 36. Test if a new radar pattern position is desired and if the production rules are not controlling the radar pattern. If true, go to Block 37. If false, return.

Block 37. Set the time that the scan pattern was shifted using the Markov model equal to the current time (Markov shift time).

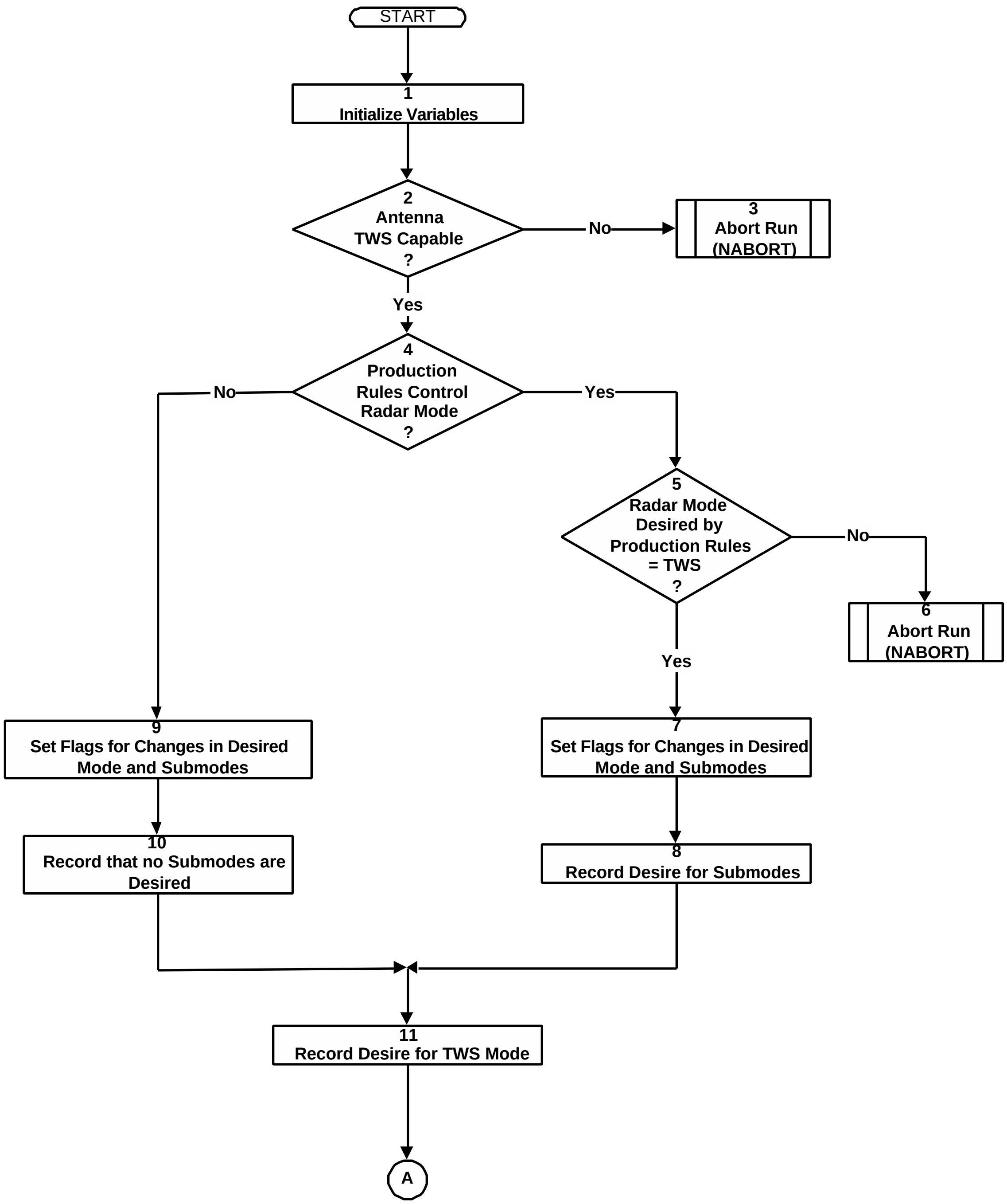

FIGURE 2.22-6. DESTWS Functional Flow Diagram (Page 1 of 3).

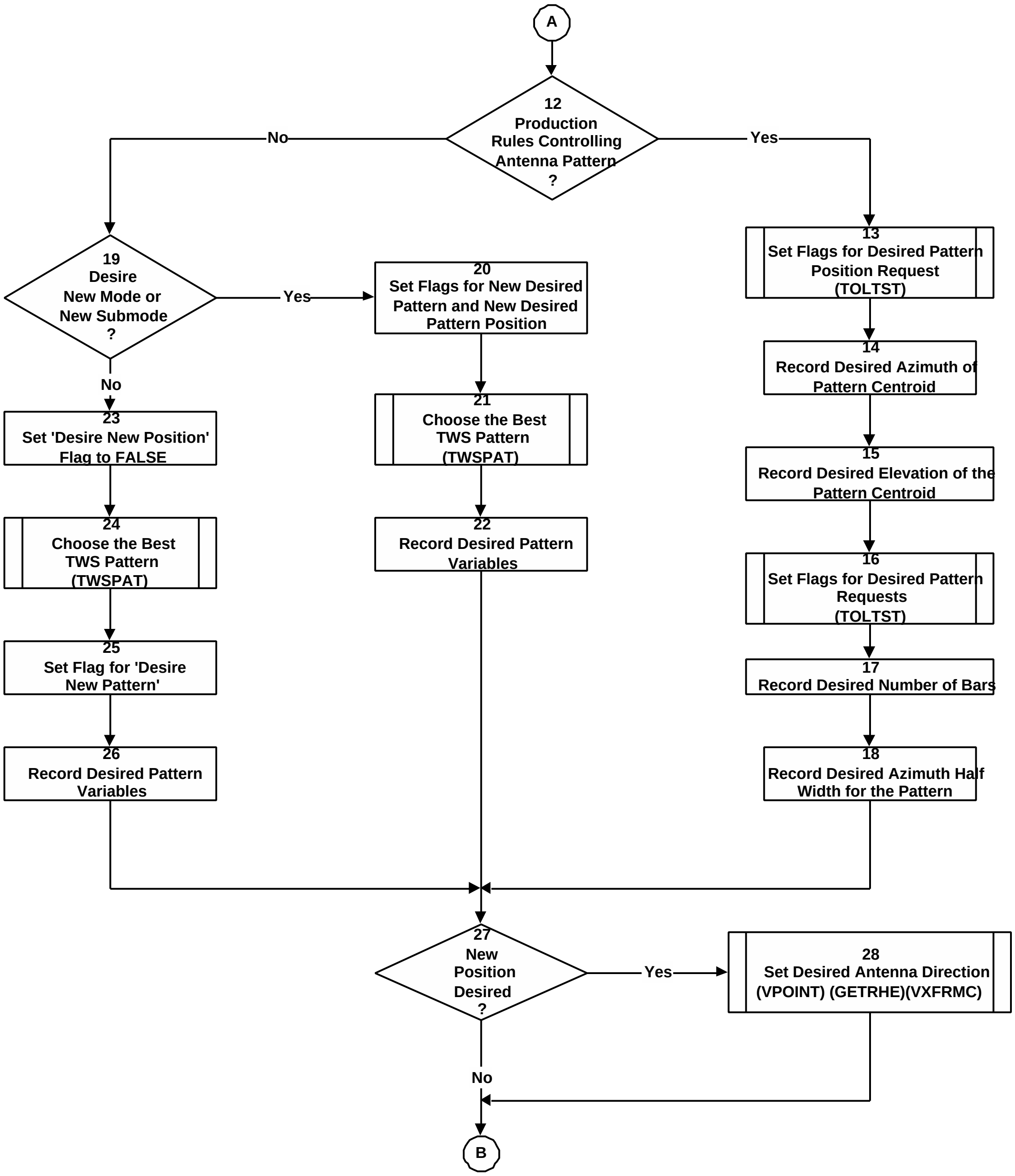

FIGURE 2.22-6. DESTWS Functional Flow Diagram (Page 2 of 3).

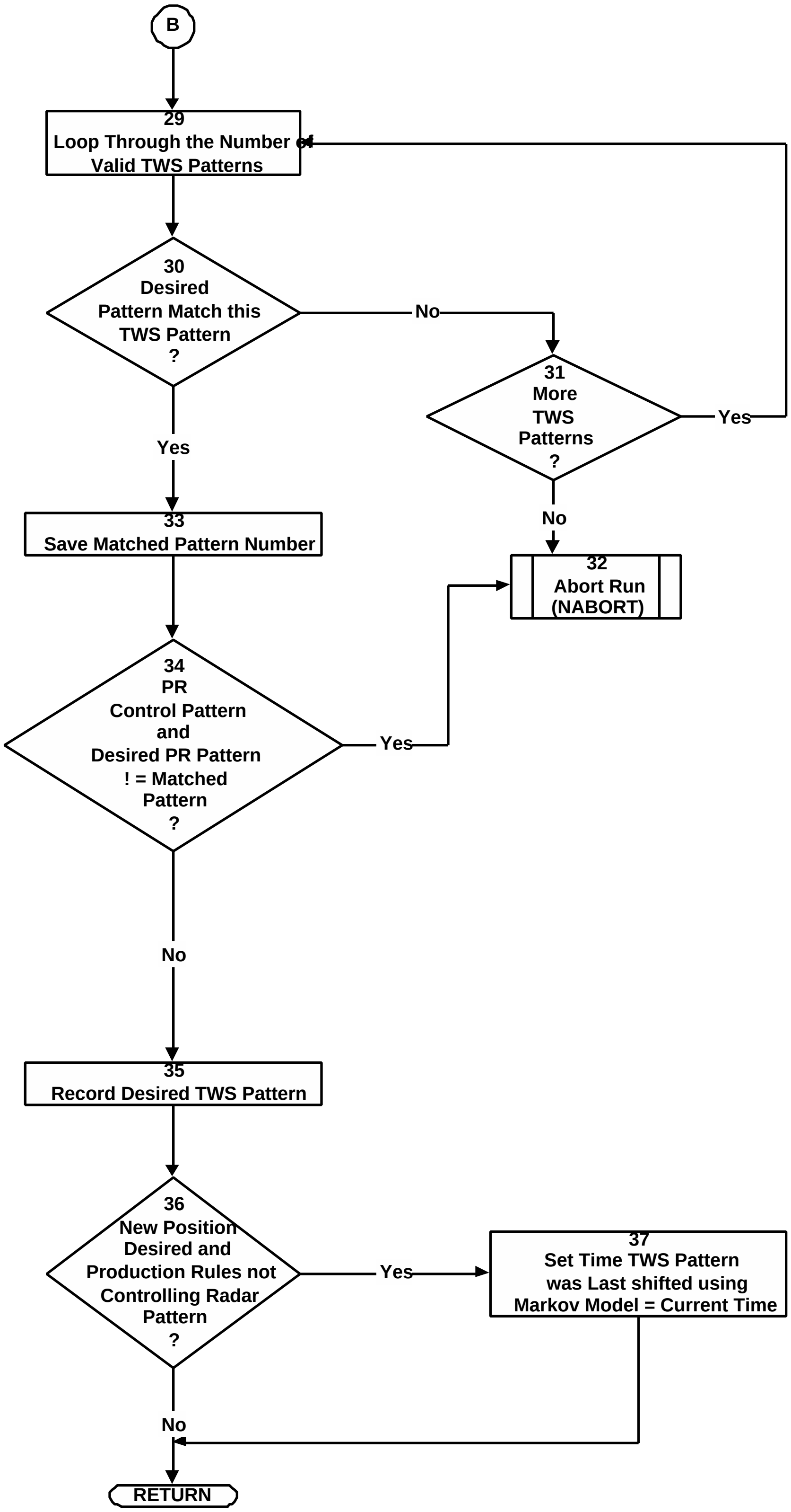

FIGURE 2.22-6. DESTWS Functional Flow Diagram (Page 3 of 3).

Subroutine DESSTT

Subroutine *desstt* records the desire for the radar to lock up on a specified target. Figure 2.22-7 is the functional flow diagram that describes the logic used to implement *desstt*. The blocks are numbered for ease of reference in the following discussion.

Block 1. Test if the intended target is on the conscious pilot's target list. If not, go to Block 2. If so, go to Block 3.

Block 2. The intended target is added to the conscious pilot's target list.

Block 3. Test if STT on the intended target is already desired. If true, go to Block 4. If false, go to Block 5.

Block 4. The 'new mode requested' flag is set to false.

Block 5. The 'new mode requested' flag is set to true.

Block 6. Record the pilot's desire for STT mode.

Block 7. Set the desire to lock flag to true and the 'lock target' to the intended target's ID. Both are variables in the pilot's mental model.

Block 8. Test if there has been a change in the desired radar mode ('new mode requested' flag is set to true). If true, go to Block 9. If false, return.

Block 9. Set the time the pilot desired lock (a mental model variable) to the current time. Set the earliest time that lock can be achieved to the current time plus *dtloc*, the average time required for this antenna to achieve lock.

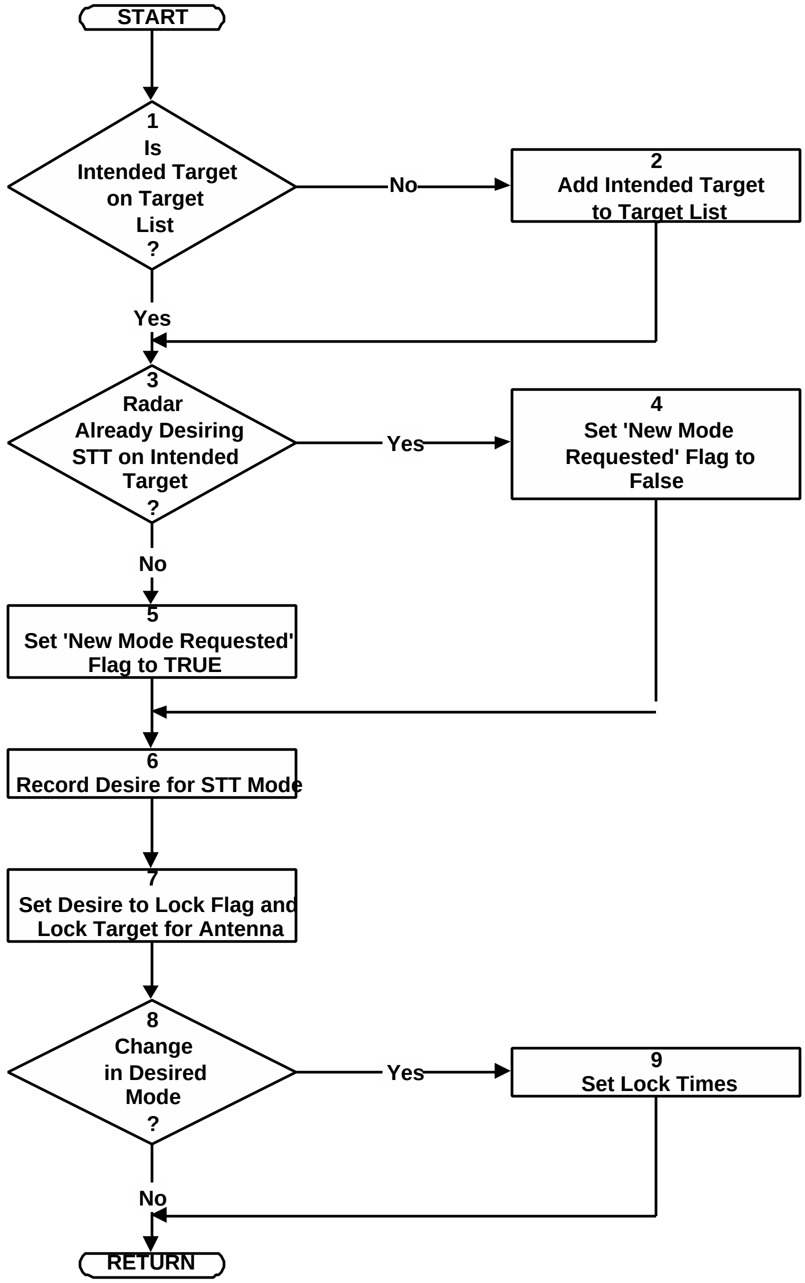

FIGURE 2.22-7. DESSTT Functional Flow Diagram.

Subroutine RDRELV

Subroutine *rdrelv* is responsible for determining the desired position of an aircraft radar antenna scan pattern in azimuth and elevation. Figure 2.22-8 is the functional flow diagram that describes the logic used to implement *rdrelv*. The blocks are numbered for ease of reference in the following discussion.

Different logic is used to position the pattern depending upon the desired radar mode. Blocks 6-28 are used to position the pattern for scan mode. Blocks 30-37 are used to position the pattern for STT mode. Blocks 38-64 are used to position the pattern for TWS/SPOT mode.

Block 1. Test if the conscious entity is a SAM site. If true, do no further processing in this subroutine and return control to the calling routine. Otherwise, position the radar antenna based on the radar mode.

Block 2. Subroutines *grdrc* and *grdrs* are called to retrieve the radar characteristics and status data for the conscious pilot's radar.

Block 3. Test if the desired radar mode is 'OFF'. If true, return control to the calling routine.

Block 4. The current desired azimuth and elevation are saved into 'old' azimuth and 'old' elevation variables to later test against the new desired azimuth and elevation in order to determine if a significant change has occurred in the desired values.

Block 5. This block is a computed go to on the desired radar mode. If desired mode is scan, go to Block 6. If STT, go to Block 30. If TWS or SPOT, go to Block 40.

Block 6. Set the 'pilot control' flag to true. A value of true indicates that the pilot is controlling radar pattern positioning. This includes production rule control. False indicates that the radar is automatically controlling positioning, as with STT lock or TWS auto mode.

Block 7. Test if the interactive pilot is controlling pattern azimuth and elevation. If true, go to Block 8. If false, go to Block 11.

Block 8. Subroutine *rdrlos* is called to convert the desired radar pattern azimuth and elevation into an earth frame LOS vector.

Block 9. Subroutine *vxfrmc* is called to convert the earth frame LOS vector to the heading frame.

Block 10. The heading frame azimuth and elevation angles are calculated from the LOS vector. Proceed to block 27 to determine if the new pattern position is significantly different from the old position.

Block 11. Test if the production rules control the radar pattern azimuth and elevation. If true, go to Block 12. Otherwise, go to Block 13.

Block 12. The desired radar pattern azimuth and elevation angles are set to the values provided by the production rules. Desired antenna pattern position in earth coordinates is then computed via a call to subroutine *rdrlos*.

Block 13. Test if the conscious pilot has selected a target. If true, go to Block 14. If false, go to Block 15.

Block 14. Subroutine *rdr_cen_tgt* is called to attempt to center the radar pattern over a particular target. For the first two frame times the radar is centered on the pilot's estimate of the target position. After that time, if the target has not been detected by the radar, the pilot will periodically move the pattern around the estimated target location in an effort to find it.

Block 15. Test if the pilot is following GCI vectors. If true, go to Block 16. If false, go to Block 23.

Block 16. The azimuth and elevation to the hostiles is calculated using geometry and the GCI information (bearing, altitude and range to hostiles).

Block 17. Test if the calculated elevation angle to the hostiles is undefined. This may happen if the conscious pilot is too close to the target, since we are mixing geometry sent to us from GCI, that may have errors associated with it, with the pilot's perception of position, that may also have errors associated with it. If true, go to Block 18. If false, go to Block 20.

Block 18. Subroutine *markov* is called to determine if a pattern shift is in order. If it has been an average of 45 seconds since the last shift, the pattern will change, i.e. the pilot will shift the center of the pattern by ±30 degrees in azimuth and ± one pattern height in elevation. When a shift is made outside of the "assigned" region, then the next time the pattern is shifted, the pilot will return the scan to the assigned region 60% of the time.

Block 19. Subroutine *rdrlos* is called to calculate the LOS vector in earth frame from azimuth and elevation in the heading frame.

Block 20. The earth frame LOS vector is calculated from the GCI information using the azimuth and elevation angles computed in Block 16.

Block 21. Subroutine *vxfrmc* is called to convert the earth frame LOS vector to the heading frame.

Block 22. The heading frame azimuth and elevation angles are calculated from the LOS vector. Proceed to block 27 to determine if the new pattern position is significantly different from the old position.

Block 23. Pilot is not following GCI. Test if the radar just dropped back into scan mode. If true, go to Block 24. Otherwise, go to Block 25.

Block 24. The radar just dropped back into scan, so reset the markov shift timer to keep from immediately moving the pattern.

Block 25. Subroutine *markov* is called to determine if a pattern shift is in order. If it has been an average of 45 seconds since the last shift, the pattern will change, i.e. the pilot will shift the center of the pattern by ±30 degrees in azimuth and ± one pattern height in elevation. When a shift is made outside of the "assigned" region, then the next time it is shifted, the pilot will return the scan to the assigned region 60% of the time.

Block 26. Subroutine *rdrlos* is called to calculate the LOS vector in earth frame from azimuth and elevation angles in the heading frame.

Block 27. Test if the production rules control the radar pattern position. If the production rules do not control the pattern position, go to Block 28. Otherwise, go to Block 67.

Block 28. The new desired radar pattern position is compared to the old position (saved in block 4). If the new pattern center azimuth does not change by more than 2 degrees or the elevation does not change by more than a half bar width, then set the desired radar pattern position azimuth and elevation equal to the old (saved) values.

Block 30. This begins the section dealing with STT mode. Test if the radar is already locked on the target. If true, go to Block 31. If false, go to Block 33.

Block 31. Set the ‘pilot control’ flag to false. This indicates that the radar is automatically controlling pattern positioning.

Block 32. The desired radar pattern center azimuth and elevation are set equal to the current values. If we have lock, we don’t want to move the radar pattern.

Block 33. Set the ‘pilot control’ flag to true. A value of true indicates that the pilot is controlling the radar pattern positioning.

Block 34. Test if CIC is the desired submode. If true, go to Block 35. Otherwise, go to Block 39.

Block 35. Subroutine *cicang* is called to calculate the radar pattern azimuth and elevation in the CIC reference frame.

There is an error in this subroutine call. The elevation argument returned by the call to cicang should have a negative sign.

Block 36. Subroutine *rdrlos* is called to calculate the LOS vector in earth frame from azimuth and elevation in the CIC frame.

Block 37. Subroutine *vxfrmc* is called to convert the earth frame LOS vector to the heading frame.

Block 38. The heading frame radar pattern azimuth and elevation angles are calculated from the heading frame LOS vector.

Block 39. Subroutine *rdr_cen_tgt* is called to try to center the radar pattern over a particular target. For the first two frame times, the radar pattern is centered on the pilot’s estimate of the target position. After that time, if the target has not been “seen” by the radar, the pilot will shift the position of the pattern around the estimated target location. This ends the STT block. Jump to Block 67.

Block 40. This begins the section dealing with TWS mode. Test if the pilot has selected a target. If not go to Block 41. If so, go to Bloc, 42.

Block 41. Subroutine *asstgt* is called to retrieve the assigned target from the pilot’s flight leader, if there is one.

Blocks 42-50 determine if the pilot wants TWS in manual or automatic mode. Manual mode will be selected if 1) The pilot has selected a target for which there is no radar track and 2) the pilot has no missiles in the air being supported by existing tracks. The motivation for test 2 is to avoid taking manual control and placing the pattern on the selected target with the risk of losing tracks supporting existing missiles.

Block 42. Initialize the 'desire TWS manual mode' flag to false.

Block 43. Test if the pilot has a selected target. If true, go to Block 44. If false, go to Block 51.

Block 44. Subroutine *anytrk* is called to retrieve tracks on the selected target.

Block 45. Test if any tracks exist on the selected target. If yes, go to Block 51. If no, go to Block 46 to check for tracks that are supporting missiles already in the air.

Block 46. This block represents the top of a DO loop over the number of missiles requiring illumination.

Block 47. Subroutine *anytrk* is called to get tracks on missile requiring illumination.

Block 48. Test if any radar tracks exist on the target of the missile requiring illumination. If any tracks exist, do not switch to manual mode (might lose tracks supporting missiles), jump to Block 51. If no tracks exist, then process the next missile on the list.

Block 49. Test if more missiles requiring illumination exist. If true, proceed to the top of the DO loop (block 46) and process the next missile. If false, go to Block 50.

Block 50. The 'desire TWS manual mode' flag is set to true, since the selected target has no radar track and there are no missiles in the air being supported by existing tracks.

Block 51. Test if the production rules control the radar pattern. If true, go to Block 52. If false, go to Block 55.

Block 52. The desired radar pattern center azimuth and elevation are set equal to the values supplied by the production rules.

Block 53. Subroutine *rdrlos* is called to convert the desired azimuth and elevation into an earth frame LOS vector.

Block 54. The 'pilot control flag' is set equal to true. Jump to Block 67.

Block 55. Test if the current TWS status is manual mode. If true, go to Block 56. If false, go to Block 62.

Block 56. Test if the mode that the antenna was in at the end of the last consciousness event was *not* TWS mode. If true (want to transition to TWS), go to Block 57. If not (already in TWS and want to stay there), go to Block 58.

Block 57. The Markov shift timer is reset to the current time to keep from immediately moving the pattern (it just came back into TWS mode).

Block 58. The 'pilot control flag' is set equal to true (we are in manual mode).

Block 59. Test if the conscious pilot has selected a target. If true, go to Block 60. If false, go to Block 61.

Block 60. Subroutine *rdr_cen_tgt* is called in an attempt to center the radar pattern over a particular target. For the first two frame times, the radar is centered on the pilot's estimate of the target position. After that time, if the target has not been "seen" by the radar, the pilot will shift the pattern around the estimated target location.

Block 61. Subroutine *markov* is called to determine if a random pattern shift is in order. If it has been an average of 45 seconds since the last shift, the pattern will change, i.e. the pilot will shift the center of the pattern by ±30 degrees in azimuth and ± one pattern height in elevation. When a shift is made outside of the "assigned" region, then the next time the pattern is shifted, the pilot will return the scan to the assigned region 60% of the time. Jump to Block 66.

Block 62. Test if the current TWS status is automatic mode. If true, go to Block 64. If false, go to Block 63.

Block 63. Subroutine *nabort* is called to terminate the run and print diagnostic messages. This block is reached if the current TWS status is neither manual or automatic, it must be one of these.

Block 64. The 'pilot control flag' is set equal to false (we are in automatic mode).

Block 65. The desired radar pattern center azimuth and elevation are set equal to the values saved at the beginning of this routine (block 4).

Block 66. Subroutine *rdrlos* is called to convert the desired azimuth and elevation into an earth frame LOS vector.

Block 67. Test if the desired radar mode is TWS or SPOT and if the current manual/automatic mode is different from the desired mode. If both conditions are true, go to Block 68. If false, go to Block 69.

Block 68. The 'desired position changed' flag is set equal to true.

Block 69. Test if the old desired pattern position is different from the new desired pattern position and the pilot controls the radar. If both conditions are true, go to Block 70. Otherwise, go to Block 71.

Block 70. The 'desired position changed' flag is set equal to true.

Block 71. The 'desired position changed' flag is set equal to false.

Block 72. The desired radar pattern centroid azimuth and elevation angles are recorded in the mental model variables *radzd* and *rdelvd*.

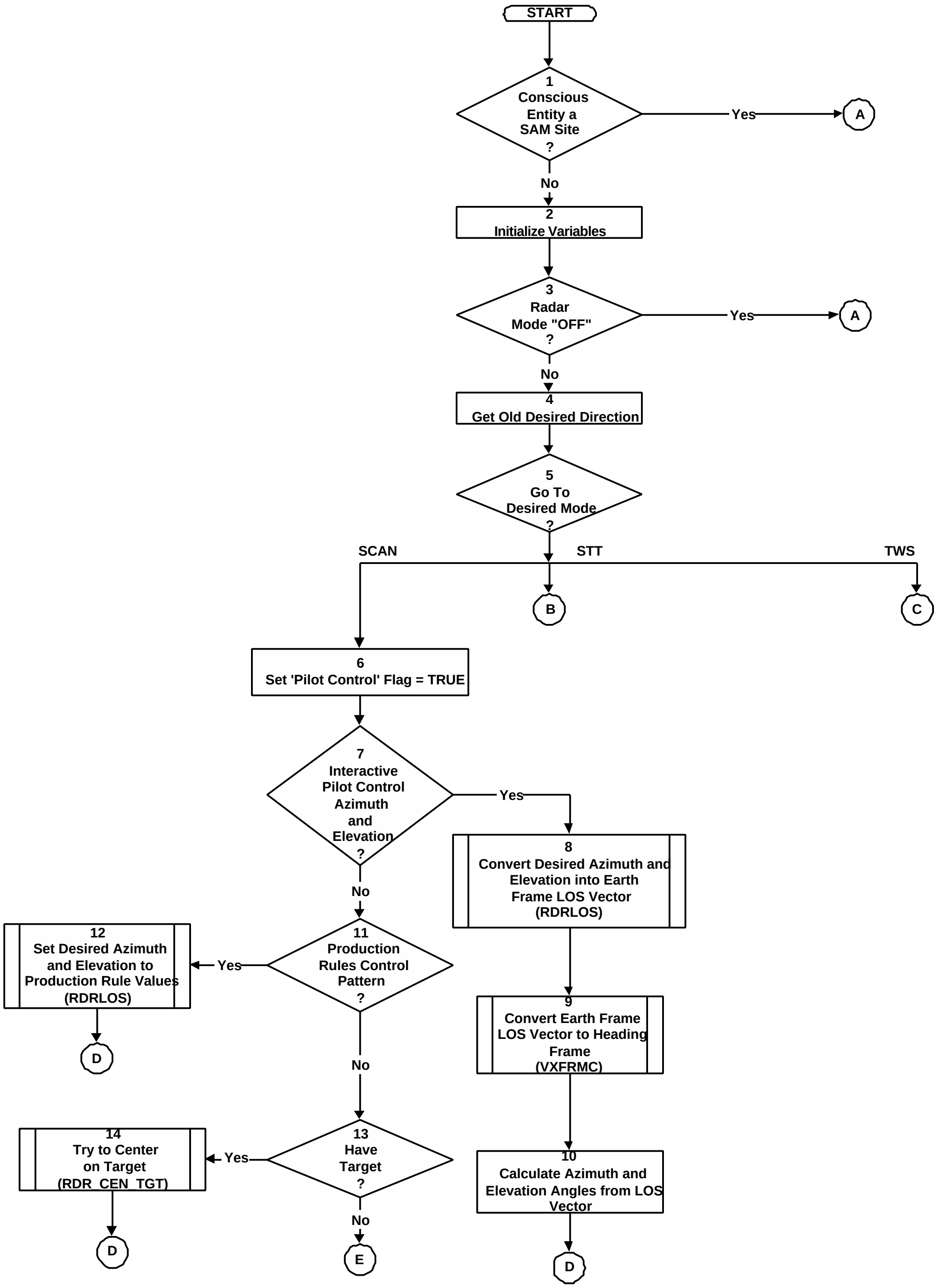

FIGURE 2.22-8. RDRELV Functional Flow Diagram (Page 1 of 6).

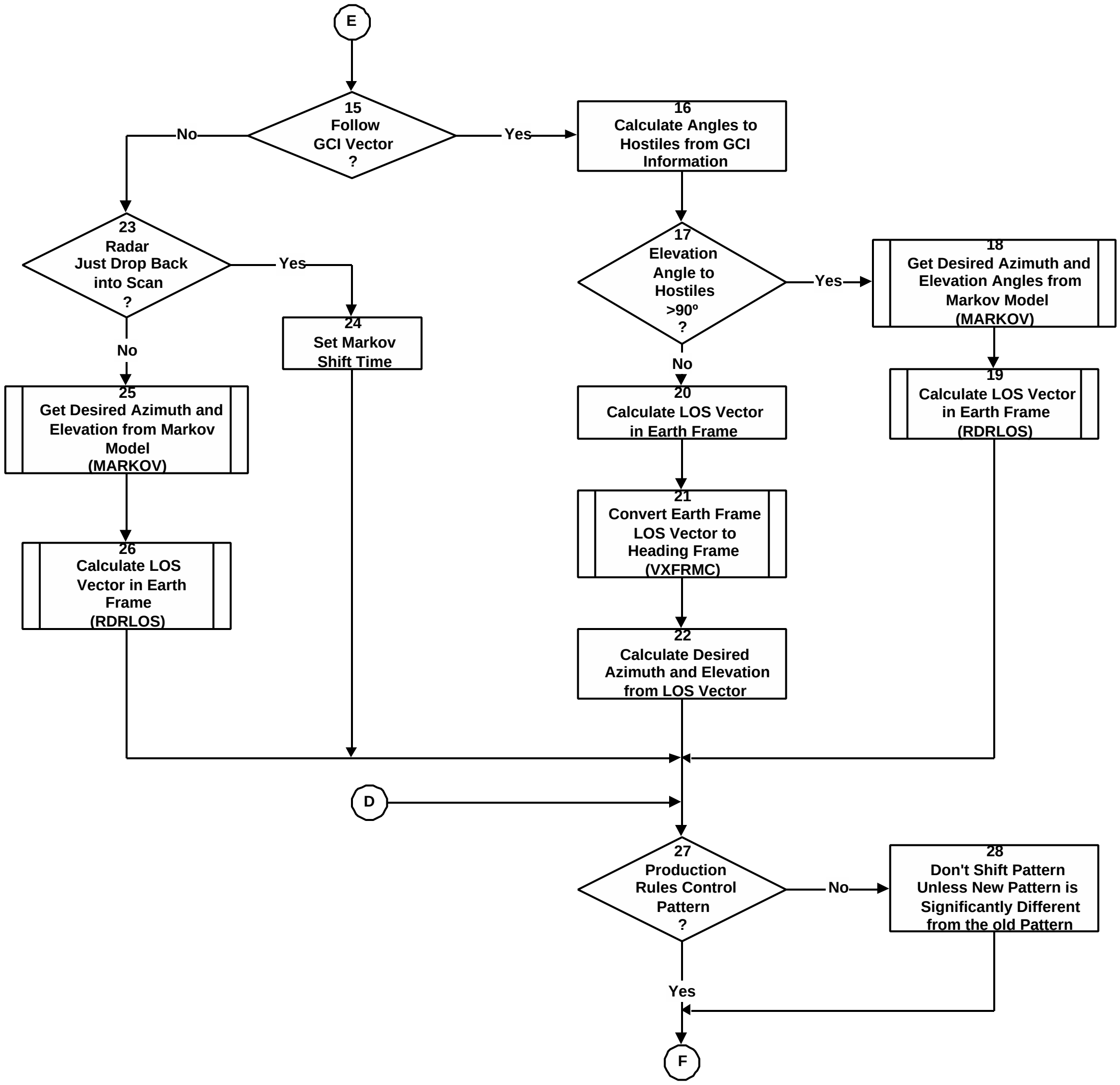

FIGURE 2.22-8. RDRELV Functional Flow Diagram (Page 2 of 6).

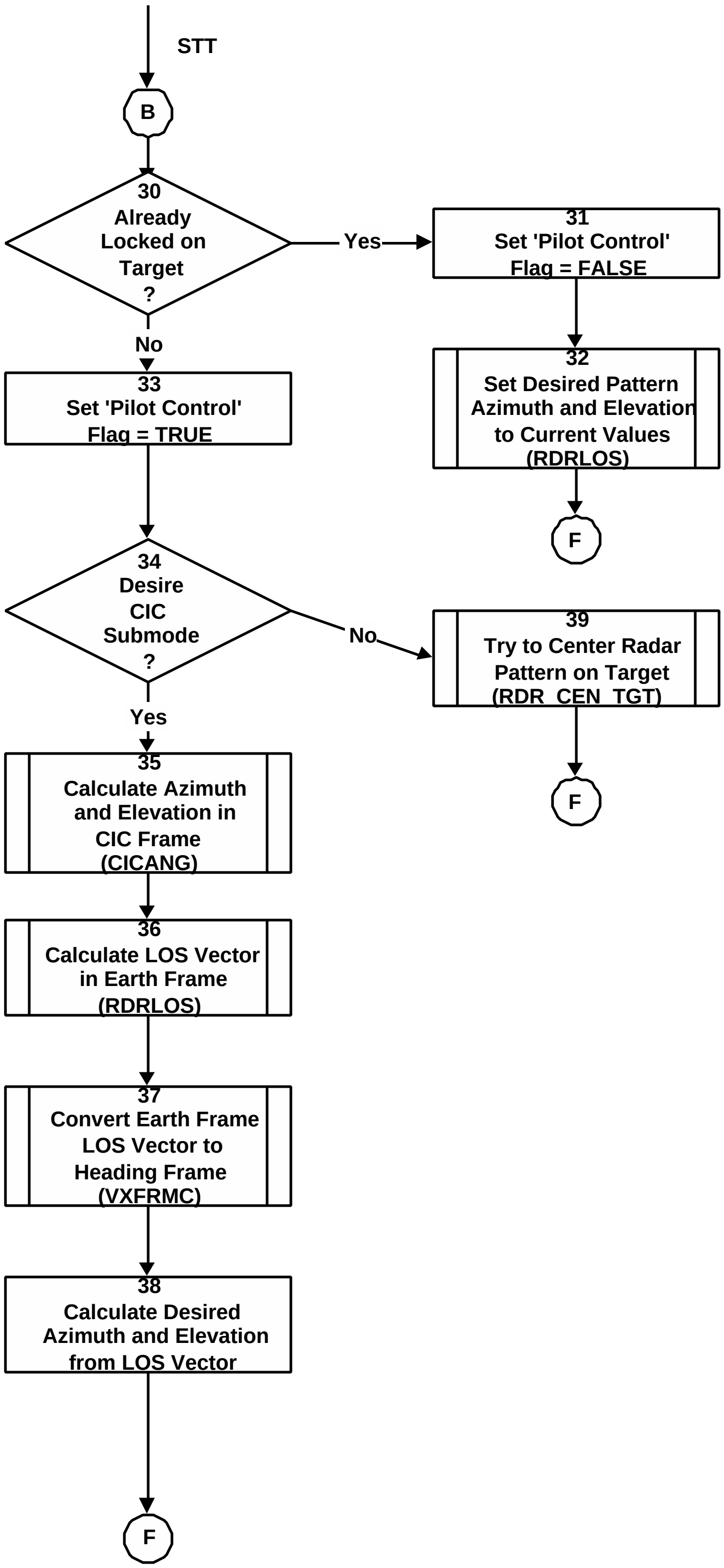

FIGURE 2.22-8. RDRELV Functional Flow Diagram (Page 3 of 6).

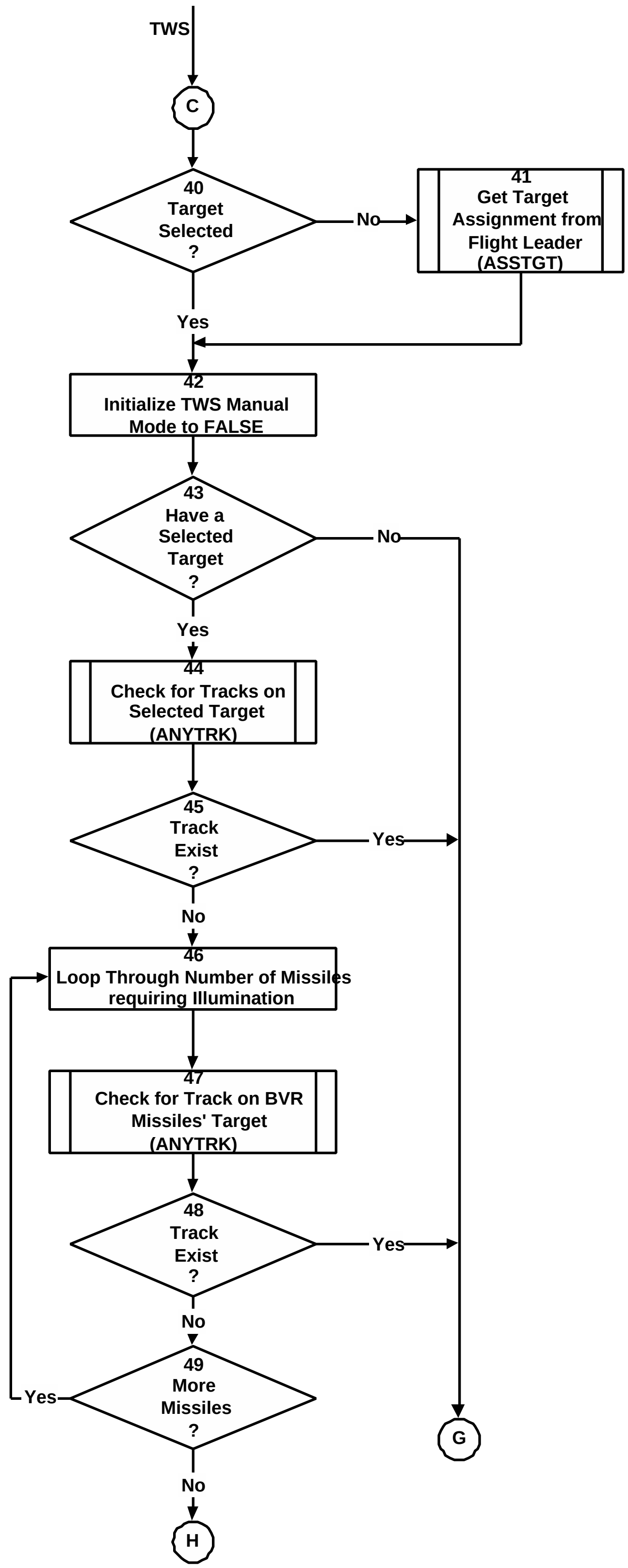

FIGURE 2.22-8. RDRELV Functional Flow Diagram (Page 4 of 6).

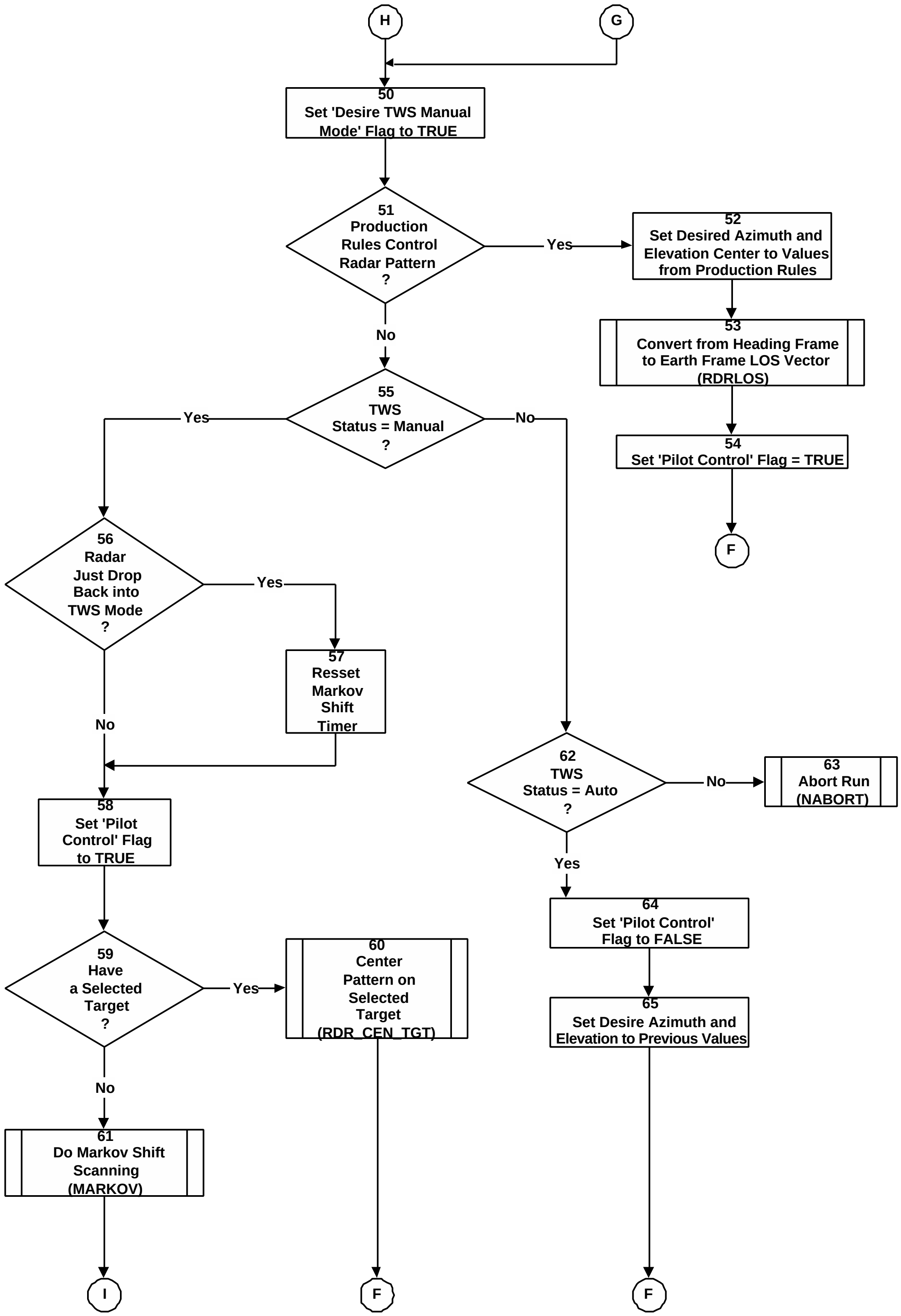

FIGURE 2.22-8. RDRELV Functional Flow Diagram (Page 5 of 6).

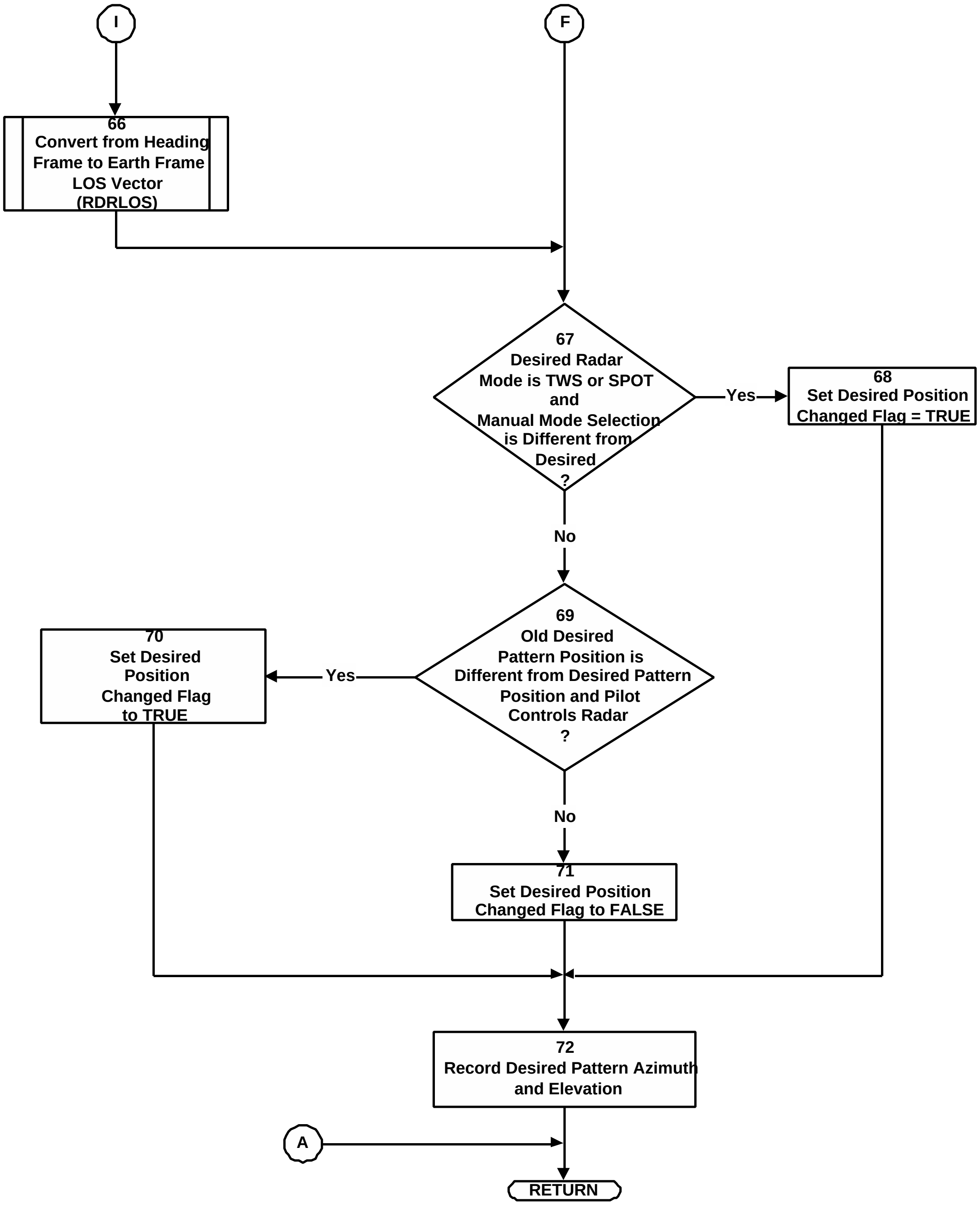

FIGURE 2.22-8. RDRELV Functional Flow Diagram (Page 6 of 6).

2.22.4 Assumptions and Limitations

The pilot decision model only controls primary radar antennas. Radar mode, pattern and pattern position for secondary antennas can only be changed through production rules.

The pilot decision model cannot turn on or off a radar device. A radar device can only be turned on or off through production rules.

A switchology delay is modeled for switching to STT radar mode. A switchology delay is not modeled for scan or TWS radar modes.

The scan pattern centroid is not shifted unless a significant difference exists (±2 degrees in azimuth, half bar width in elevation) between the desired pattern position and the current pattern position.

TWS antennas may have up to 3 different scanning patterns. It is assumed that no two patterns will have both the same number of bars and the same azimuth half width.

A TWS capable antenna may also have multi-target track (MTT) capability. MTT capability is only available for electronically scanned antennas.

CIC submode may only be entered through the use of production rules.

Velocity search submode may only be entered through the use of production rules.

Burst submode is only available for electronically scanned antennas. It may only be entered through the use of production rules.

2.22.5 Known Problems or Anomalies

No distinction is made between command guided missiles supported by the radar and command guided missiles supported by other avionics devices. This may bias the pilot toward STT when it is not necessary.

Code still exists in the radar control routines to implement interactive pilot control, even though this feature is now obsolete and not functional. This has no effect upon model function, but may be confusing to anyone reading the code.

In subroutine *rdrelv*, there is a sign error in the code that determines the desired pattern elevation when STT and CIC are the desired mode and submode. The effect of this will be an error in the elevation of the position of the CIC scan pattern. Since CIC mode causes the radar to scan vertically, and the scan patterns are generally much closer vertically than horizontally, this error is only expected to cause a problem if the difference in elevation between the perceived and actual target location is large. Otherwise, the radar scan pattern will still sweep over the target.

A desired scan pattern position is defined when the pilot desires to lock up in STT, but this is not used by the radar hardware model, which assumes that the STT pattern position is centered well enough that the scan will cover the target.